1991年3月1日発行（毎月1回1日発行）第10巻3号通巻107号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集 MIDI&MUSIC PROCESSING

SX-WINDOWプログラミング SXLIFE完結編

周辺機器紹介 カラープリンタ/内蔵ハードディスク

CARD PRO-68K Ver.2.0/Musicstudio PRO-68K Ver.2.0

1991

オー/エックス

定価560円

SHARP

X68000 SUPER登場

このたび新たにラインアップされた“X68000 SUPER”は、すでに発売されている“SUPER HD”と同様、SCSIインターフェイスを標準装備しています。また、その他のシリーズにはオプションとしてSCSIボード(CZ-6BS1)がサポートされ、大容量外部記憶装置をはじめ、各種SCSI装置との接続が可能になったのは、ご存じのとおりです。

SCSI規格とは……

SCSIは1986年にANSI(米国規格協会)で規格化された仕様で、Small Computer System Interfaceの略。小型コンピュータの周辺機器接続のための世界共通の規格です。大容量外部記憶装置(大容量ハードディスク、CD-ROM、DATなど)に加え、登場が期待される高速スキャナ、次世代プリンタなどのSCSI装置を、デイジーチェーン方式で最大7台まで接続可能。大容量データの高速転送、および単一のインターフェイスでの周辺機器の複数制御が特長です。

X68000と大容量メディア

サウンドクリエーション&コンピュータグラフィックス。X68000のオハコともいうべきこの領域は、感性あふれるユーザーにとって最も魅力的である反面、表現の繊細さに比例して必要な外部記憶容量も増大します。サンプリング、MIDI、レイトレ……。その潜在能力をフルに引き出すには、大容量メディアへの対応が必須です。たとえば、新発売の光磁気ディスク(CZ-6MO1)と光磁気ディスクカートリッジ(JY-701MPA)なら、ディスク1枚で65,536色画像にして1,000枚強、15.6kHzの音声サンプリングデータで約20時間強もの情報を記憶できます。絵に書いた餅とされていた「画像データベース」も、「AD PCMデータライブラリ」さえも、も実用レベル。SCSIの採用が、夢の大容量メディアに応えてくれるからです。

▲MOディスクドライブ情報　▲MOドライブディレクトリ情報　▲MOディスクのフォーマット

●デイジーチェーン方式による接続例

SCSIボードCZ-6BS1▶
標準価格29,800円(税別)

X68000の先見性

初代X68000は、すでにハードディスクインターフェイスを内蔵していたこと。当時また一般的ではなかったハードディスクに対して先見の発想で臨んでいたわけです。今回のSCSI対応も同様、100MBを超える大容量メディアハンドリングがスタンダードになる日も、そう遠くはありません。

大容量ハードディスクか？光磁気ディスクか？それとも……

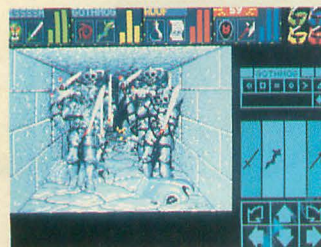
考えもしなかった新しいデバイスか。新製品X68000SUPERのSCSIインターフェイスに何を接続するかは、賢明なユーザー諸兄にお任せするとして…。このマシンがまた新たな一歩を踏み出したことに異論はないはずです。蛇足ながらこのSUPERシリーズに関していわせてもらえれば、その日から大容量ハンドリングをお望みの方にはSUPER HDを。未来に夢を託したユーザーはSUPERといったところでしょうか。

＊SCSI装置をご使用の場合は、Human68k Ver2.0以上でご使用ください。
＊ビジュアルシェル上からはSCSI装置はご使用になれません。

特集　MIDI & MUSIC PROCESSING

C-TRACE68+

(で)のショートプロぱーてぃ

アトミック・ロボキッド

続ダンジョン・マスター カオスの逆襲

新製品 CZ-8PC5

Oh!X

C O N T

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／岡崎栄子 浅井研二 ●協力／有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／グループごじら

表紙絵：塚田 哲也

E N T S

1991 MAR.
3

■広告目次

SHARP

システムパフォーマンスを実証する多彩なペリフェラル。

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格99,800円(税別)
(チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-613D-TN・-BK・-GY
標準価格135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格19,800円(税別)
(14/15型用)

カラーディスプレイ

14型カラーディスプレイ
CZ-606D-TN・BK・-GY
標準価格79,800円(税別)
(チルトスタンド標準装備)

14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格148,000円(税別)
(スピーカー2個同梱)

チューナー

RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格33,100円(税別)
(リモコン付)

アートツール

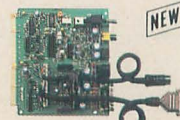
画像入力

カラーイメージスキャナ※1
CZ-8NS1
標準価格188,000円(税別)

スキャナ用パラレルボード
CZ-6BN1
標準価格29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格69,800円(税別)

映像出力

ビデオボード※3
CZ-6BV1
標準価格21,000円(税別)

プリンタ

熱転写カラープリンタ

48ドット
熱転写カラー漢字プリンタ
★CZ-8PC4
CZ-8PC4-GY
標準価格99,800円(税別)
(信号ケーブル同梱)

48ドット
熱転写カラー漢字プリンタ
CZ-8PC5-BK
標準価格96,800円(税別)
(信号ケーブル同梱)

カラードットプリンタ

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格130,000円(税別)
(信号ケーブル同梱)

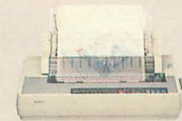
24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格160,000円(税別)
(信号ケーブル同梱)

カラービデオプリンタ

カラービデオプリンタ
CZ-6PV1
標準価格198,000円(税別)
(信号ケーブル同梱)

ドットプリンタ

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格97,800円(税別)
(信号ケーブル同梱)

カラーイメージジェット

カラーイメージジェット※4
IO-735X
標準価格248,000円(税別)
(信号ケーブル別売)

ファイル

光磁気ディスク

光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格450,000円(税別
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク

増設用ハードディスクドライブ(40MB)
(CZ-602C/603C/652C/653C内蔵用)
CZ-64H※
標準価格120,000円(税別
(取付費別

増設用ハードディスクドライブ(80MB)
(CZ-604C内蔵用)
CZ-68H※
標準価格160,000円(税別
(取付費別
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

ハードディスクユニット(20MB)
CZ-620H
標準価格178,000円(税別

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1 標準価格29,800円(税別)で接続してください。 ※2 CZ-603D、604D、606D、CU-21HDをご使用の場合は、RGBシステムチューナー CZ-6TU(別売)が必要です。 ※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。 ※4 別売の信号ケーブル IO-73CX 標準価格5,500円(税別)で接続してください。 ※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cにご使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。(但し、CZ-604C、623Cは不要)また、X68000用OS Human68K ver.2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大

資料請求券
X68000ペリフェラル
Oh!X
3条

SHARP

ハイアビリティを実証する多彩なソフトウェア。

ドロー編集、WYSIWYG印刷、こんなC.G.ツールが欲しかった。

本格的なロゴタイプやPOPを簡単に作成できるグラフィックツールです。優先順位が任意に指定できるドローセル、ペイントセル、テキストセルの3つの仮想セルで、目的にあった自由なグラフィックが駆使できます。また印刷は、画面イメージがそのまま印刷イメージとなるWYSIWYG(What You See Is What You Get)を実現。A6/A5/A4/A3/B6/B5/B4/葉書サイズで8色カラー印字できます。

〈ドローセル〉ベジェ曲線によって少ないデータ量でも複雑な絵を描くことができます。エンベロープ変形を始めとした豊富な編集機能を持っており、拡大、縮小しても絵の美しさは変わりません。またテキストセルで作成したベクトルフォントデータを自由に変形し、オリジナルロゴタイプやPOPを作成できます。

〈ペイントセル〉ペンやエアーブラシ、ペンキなどを使って、ピクセルで構成されたビットマップ図形を描くことができます。また、「NEW PrintShop PRO-68K」や「X-BASIC」、「Z's STAFF PRO-68K」のデータ取り込みやイメージスキャナによる取り込みをサポートしています。

〈テキストセル〉通常の文字入力機能に加え、ベースライン変形などの多彩な編集機能によって自由に文字の加工ができます。また英数文字のベクトルフォントを標準装備。さらに「Z's STAFF PRO-68K Ver2.0」、「書体倶楽部」の日本語ベクトルフォントが利用可能。また、内蔵の漢字ROMフォントも自動的にベクトルフォントデータに変換しますので、簡単に日本語ロゴタイプを作成することができます。

※「Z's STAFF PRO-68K」、「書体倶楽部」は、㈱Zeitの製品です。
※本ソフトの動作には、メインメモリ2MBが必要です。

CANVAS PRO-68K CZ-249GS 標準価格29,800円(税別)

●主として個人用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

海のデータ/動物のデータ/スポーツのデータ/鳥のデータ/人物のデータ/食物のデータ/昆虫のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.1
CZ-255GS 標準価格8,800円(税別)

●主としてビジネス用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

OA関係のデータ/飾りのデータ/コンピュータ関係のデータ/POPのデータ/国旗のデータ/字体のデータ/地図のデータ/乗り物のデータ

CANVAS PRO-68K ドローグラフィックライブラリ VOL.2
CZ-256GS 標準価格8,800円(税別)

バージョンアップされたCコンパイラと、強力なBASTOCチェッカー。

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできる「ソースコードデバッガ」を搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(830種以上)となっています。C言語の標準であるANSI規格準拠をさらに強化。「プログラム保守ユーティリティ(MAKE)」や「ライブラリアン」など各種ツールを追加しました。その他「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「コンバータ」、などのツールが装備されています。

※C compiler PRO-68K(CZ-211LS)を既にお持ちの方は、登録カードをもとに有償バージョンアップを行います。
※本ソフトの動作にはメインメモリ2MBが必要です。

CZ-245LS 標準価格44,800円(税別)

トラブルエラーの悩み解消!「XBAStoC」の強力ツールの登場です。

X-BASICプログラムのコンパイル時、発見しづらいトラブルエラーに悩まされていたプログラムの問題点をひとつひとつ指摘。エラーとなる直接原因だけでなく、注意項目も指摘します。これにより、X-BASICでは実行できたのにコンパイルするとエラーが発生する、といったプログラムの修正が簡単にできます。

●指摘したトラブルの結果を、画面やプリンタなどの外部デバイスに簡単に出力できます。●エラーラインとエラーレポート、2つのエラーファイルを自動的に生成。●グラフィカルな画面による簡単操作。●コマンドラインからダイレクトに操作を指定。バッチファイルに組み込むなどの修正作業の自動化が可能。●GP-IBボード(CZ-6BG1)とユニバーサルI/Oボード(CZ-6BU1)付属の拡張外部関数もコンパイル可能。

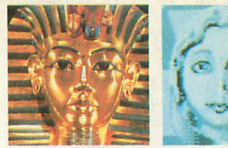
※X-BASICプログラムをコンパイルするためには、別売の「C compiler PRO-68K」(CZ-211LS)または「C compiler PRO-68K ver2.0」(CZ-245LS)が必要です。

XBAStoC CHECKER PRO-68K
CZ-260LS 標準価格9,800円(税別)

シャープ株式会社●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。

資料請求券 CZソフト Oh!X 3体

シャープオリジナルソフトウェア
X68000

お望みのワークベンチへ。

ビジネスツール

Hyperword
■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。

CYBERNOTE PRO-68K
■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-300Lが必要)。

Stationery PRO-68K
■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationeryPRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-300Lが必要)。

TOP給与計算エキスパート
■CZ-228BS標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計
■CZ-227BS標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。

グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

通信ツール

Communication PRO-68K ver2.0
■CZ-257CS 標準価格19,800円(税別)

Communication PRO-68Kのバージョンアップ版です。MNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポートしています。

※バージョンアップ対応中。

開発ツール

SX-WINDOW ver1.0
■CZ-259SS 標準価格6,800円(税別)

複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。IOCSコールを利用したソフトの処理速度を高速化するIOCS.Xを付属。

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

マルチタスク機能、リアルタイム機能を活かした使いやすく機能的なOS環境を提供します。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

システムパフォーマンスをさらに高める処理機能を付加したHuman68kの最新バージョンです。

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS標準価格188,000円(税別)

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS 標準価格28,800円(税別)

MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987

ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987

アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

アクションゲーム
〈サイバリオン〉
■CZ-229AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部サッカー編〉
■CZ-262AS
標準価格8,800円(税別)
©TECHNOS JAPAN CORP. 1990

〈お詫びと訂正〉2月号の本広告において、一部ゲームソフトの機種と画面の対応がとれていないものがありました。慎んでお詫びして訂正申し上げます。正しくは本号広告をご参照ください。

アッパレ
X68000EXEクラブ
EXEおみこし活動に参加しよう!
◉まずはEXEクラブへ◉
入会無料で3つのメリット!手続きは本体同梱の入会申込ハガキを送るだけ。
メリット1▶会員番号入りオリジナル会員証電卓がもらえます。
メリット2▶各種フェアご優待・イベント案内等、数々の特典があります。
メリット3▶X68000の活用情報が手に入る「EXEおみこし活動」に参加できます。
※「入会申込ハガキをなくしてしまった」という方は、おみこし活動隊までご電話下さい。
◉EXEおみこし活動とは?◉
いわば「X68000ユーザーの、X68000ユーザによる、X68000ユーザーのための」活動です。おみこしPRESSを通じて会員同士情報を交換し、もっと68を使いこなして盛り上がってしまおう!というワケ(モデムがなくてもできるパソコン通信のようなもの?!)なので、X68000へのラブコール、会員独自のテクニック、活用法(マニアックなものでなくても他の会員には貴重!)等あなたの68自慢をドシドシ聞かせてください。会員からのメッセージは「おみこし活動隊」が整理してコミュニケーションペーパー「おみこしPRESS」にバッチリ掲載します。
おみこし活動隊
へのアクセス方法
投稿受付/大阪市淀川区西中島1丁目9-16 新大阪ストロングビル2F
X68000EXEクラブ「おみこし活動隊」係
電話受付06-886-0354 FAX受付06-304-1539
★受付時間……平日/昼2時～夜8時
日祝/昼12時～夜8時
◉そしてEXE会員究極の◉
「おみこしかつぎ人」を大募集!
「かつぎ人」とは、より積極的におみこし活動に参加する人のこと。EXE会員は「かつぎ人」になることで、X68000ユーザーとしてますます充実、3つのメリットで強力にサポートされます。
《メリット1》「おみこしかつぎ人の集い」に参加できます。
シャープとEXE会員の双方向コミュニケーションの場として開設されるX68000情報交換会「おみこしかつぎ人の集い」は、シャープの68スタッフと直に意見交換ができるおいしいチャンス。68の最新情報満載のこの会に参加すれば、68ユーザーとしてトップレベルです。
《メリット2》「おみこしPRESS」定期送付。
お店まで足を運ばなくても「おみこしPRESS」が毎号お手元に届きます。
《メリット3》「ソフトウェア・フィールド」直送。
X68000最新ソフト・各種周辺機器が一覧できる「ソフトウェア・フィールド」を半年に1回お送りします。
おみこしかつぎ人になるには……
以下の年会費(おみこしかつぎ代)が必要です。〈個人入会〉3,000円/〈グループ入会(5人1組)〉2,500円/1人(郵便振込にて申込受付)
★詳細は店頭の「おみこしPRESS」をご覧になるか、
または「おみこし活動隊」にお電話ください。
カードデザインコミュニケーション
カードデザインコンペ作品大募集
X68000で、あなただけのとびっきりのデザインカードを…。「よし、これだ!」と思ったら、あとは応募するしかない!!ソフトは自由。点数無制限。アイデアを生かした楽しいカードデザインを募集します。A・B・C各部門毎に優秀作品を選考、オリジナルカレンダーに掲載してプレゼントします。多数の力作をお待ちしています。
《応募要領》
◎作品分類…部門A/クリスマスカード、ニューイヤーカード 部門B/バレンタインカード、バースディカード 部門C/暑中見舞カード、サークル・趣味の会お知らせカード。◎応募資格…X68000で制作されたカードに限ります。◎応募方法…ポストカードサイズ(10×15cm)の作品裏面に必要事項(作品名・部門名・名前・住所・X68000使用機種名・周辺機器名・ソフト名)を明記し、封筒に入れて郵便にてお送りください。◎受付期間…90年10月1日～91年2月28日(消印有効)。◎賞…優秀作品賞/優秀作品掲載のオリジナルカレンダー及びオリジナル表彰楯を進呈。参加賞/応募者全員に優秀作品掲載のオリジナルカレンダーを進呈。※応募作品にかかわる諸権利は主催者に帰属するものとして作品の返却は致しません。◎発表…オリジナルカレンダー・オリジナル表彰楯の発送をもってかえさせていただきます。
◆カードデザイン屋さんOPEN◆
ご来店の皆様にカードデザイン作りの実演を楽しんでいただく「カードデザイン屋さん」を開催します。グラフィックソフト「CANVAS PRO-68K」で自分だけのカードデザインにチャレンジしてください。ぜひお近くのEXEショップへ。
応募作品続々のカードデザインコンペも締切目前!!
カードデザイン屋さん通いで密かに腕を磨いていた君、健闘を祈る!!
ヨミ!
セッセ
セッセ
シャープ株式会社
●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545

'91 NEW WAVE
itec
HARD DISK UNIT
SHARP X68000専用
イテックのハードディスクがさらに進化しました。
HARP X68000専用SCSIハードディスクユニットIT X80S
T X130Sの登場です。
イテックでは早くからX68000専用ハードディスクに着手し今ま
SASIタイプハードディスクに多大なる評価を得てきました。
の技術をさらに進化させた結果次代が求めるSCSIタイプハー
ディスクの発表となったわけです。
8000 SUPER HDの発売によりSCSIのもつ拡張性能の良
があらためて認識されるようになりました。また従来のX68000
ーズにもSHARP製SCSIインターフェースボードCZ-6BS1
用いることにより完全にSCSI機器を周辺装置として接続が可
になりました。アイテックのSCSIハードディスクはこのどち
対応できるもので最大7台デージチェーンでの接続、また
の大容量80MB/130MBさらに60×120×243(mm)のコンパクト
サイズを実現。
S IT X130Sをあなたはどのように
SELECT
POWER
itec
IT X130S
SELECT
POWER
itec
IT X80S
IT X80S ¥128,000
国産高精度SCSIドライブ搭載
容量80MB
平均アクセスタイム20ms
接続ケーブル添付
IT X130S ¥158,000
国産高精度SCSIドライブ搭載
大容量130MB
平均アクセスタイム20ms
接続ケーブル添付
アイテック株式会社
本社 〒550 大阪市西区新町1丁目25番14号 TEL:(06)532-0216 FAX:(06)532-7253
関西営業所 〒550 大阪市西区新町1丁目25番14号 TEL:(06)532-0120 FAX:(06)532-0543
関東営業所 〒275 千葉県習志野市谷津1丁目12番5号 TEL:(0474)77-7564 FAX:(0474)73-2759
名古屋営業所 〒460 名古屋市中区大須2丁目28番31号 TEL:(052)212-1487 FAX:(052)212-1627
インフォメーションセンター
TEL:(06)532-0320

C-TRACE CG コンペティション'91 開催!

◎応募要項◎

応募期間 平成3年1/20〜3/20

発　表 平成3年5月発売のASCII Oh!Xの当社広告紙面

募集規定 C-TRACEユーザーがC-TRACEを使用して作成したCG静止画像、アニメーション

募集部門 ●静止画キャラクター部門●静止画アート部門●静止画産業デザイン部門●アニメーション部門

応募方法 キャストまで応募要項を請求後、静止画CG画像はフロッピーディスク、アニメーションはビデオテープ(VHS、ベータ、8㎜のいずれか)で応募

賞 ■グランプリ、準グランプリ、金賞、銀賞、銅賞(各1名)、■ステゴちゃん賞(3名)、■メーカー特別賞

賞　品 ■グランプリ、準グランプリ、金賞、銀賞、銅賞の順に以下の賞品の中から1つを選択——海外旅行クーポン、トランスピュータ、フレームバッファ、キャスト商品券(10万円券、7万円券、3万円券) ■ステゴちゃん賞——ステゴちゃんぬいぐるみ ■メーカー特別賞——ソニーコンピュータシステム賞〈データディスクマン〉、アイ・オー・データ機器賞〈EMSメモリボード(2M)〉、SHARP賞〈コプロセッサボード〉、ASCII賞〈月刊アスキー1年間無料購読〉

※応募者全員にもれなくキャストオリジナルポストカードをプレゼント

審査員(敬称略) ASCII・Oh!X・Oh!PC:各編集部、CGキッチンまざあぐうす代表 長谷川一光、C-TRACEユーザークラブ会長 小石光、東京工学院芸術専門学校講師 塩沢左千子、超人(超能力者・平成2年11/27フジTV出演) 玉手峰人、イラストレーター 伊川英雄、各協賛会社

協　賛 ソニーコンピュータシステム株式会社
株式会社アイ・オー・データ機器
シャープ株式会社
月刊アスキー編集部

メモリー解放宣言。
C-TRACE98 EXTENDER
価格¥128,000(税別)

●PC-9800シリーズ、PC-286, 386シリーズ●メインメモリとして最大16M使用可能●EMSによるメモリ拡張のようにスピードを犠牲にしません●30%の高速化(当社C-TRACE Ver.3.0比)●Ver.3.0からのステップアップ受付中〈ソフトウエア〉

1670万色表示。
フルカラーフレームバッファ★
価格¥69,800(税込)

●PC-9800シリーズ、PC-286, 386シリーズ●1670万色同時表示●フレームバッファ制御のためのサンプルソース付き●RAMディスクドライバ付き●ズーム、スクロール&パン機能をハードウェアでサポート●フレームバッファ+ユーティリティディスク(2枚)

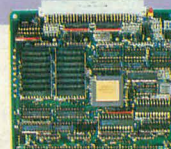
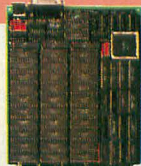
超高速。
C-TRACE TP Ver.3.0★
価格¥298,000(税込)

●PC-9800シリーズ、PC-286, 386シリーズ、X68000シリーズ●パソコンでレイトレーシングをワークステーション並みのスピードで実行可能●並列処理によりスピードアップも可能●トランスピュータボード+C-TRACE Ver.3.0のセット

メタボール対応。
C-TRACE+ NEW
価格¥198,000(税別)

●PC-9800シリーズ、PC-286, 386シリーズ、X68000シリーズ●C-TRACEシリーズ最上位のスペック●メタボール対応●スポット光源対応●αチャンネル対応により高度な合成が可能●スコープ機能対応により部分的に画像の再計算が可能●DOSエクステンダにも対応●差額交換受付中〈ソフトウエア〉

C-TRACE68 Ver.3.0	¥98,000(税別)
C-TRACE98 Ver.3.0	¥98,000(税別)
C-TRACE TOWNS	¥68,000(税別)
C-TRACE NEWS Ver.3.0	¥530,000(税別)

一部クレジット取扱可

★の製品は店頭販売いたしておりません。直接当社まで、お申し込みください。

※電話・FAX番号が変わりました。お間違えのないようお願い致します。

Cast

未来とは定められ

人類の歴史は偶然の結果の記録ではない。
それは、定められたひとつの目的にしたがって操作された結果である。
あらゆる予言の書の存在…。
なぜ人が未来を知り得るのであろうか?
人類は定められた運命を変える事ができるのか?

X68000 ONLY 5"2HD(5枚組) 価格¥12,000(税抜)

■マウスオペレーションで簡単操作　■200枚を超える美しいグラフィック
■史実の謎に迫る野心的ストーリー　■現地取材をもとにしたリアルな構成

スペキュレーティブ・アドベンチャー

シグナトリー

SIGNATORY

――調印者――

提供■NCS　制作総指揮・総監督・原作■鈴木 力　脚本■成田伸子　出演■ケニー・フィリップ/バーバラ・ドゥーディ/トーマス・スウェイジ他　制作■Tenky

■制作スタッフ■スクリプト 皆川正三 ■SE・プログラム 橋谷利幸 ■メイン・プログラム Hかずき ■チーフデザイン 石井秀明 ■デザイン 本間繁二郎 木村政幸 矢田 智 古沢雅子 ■音楽・効果音 高橋大昌 ■NY・南米取材 青空風太郎 ■NY取材協力/酒井友

オリジナルテレカ・プレゼント! 「シクナトリー」をお買い上げ頂き商品内のユーザー・ハカキをお送り下さった方の中から、先着500名様にオリジナルテレカをプレゼント。

た運命なのか?

3月29日より全国公開予定!

NCS 日本コンピュータシステム株式会社
〒106 東京都港区西麻布4-16-13 第28森ビル TEL.03-3486-6314(代表)
お問い合わせはソフトウェア プロダクト部(直通) 03-3486-6588(受付時間) 9時 18時

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

■ビッグバーゲンセール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!!

オクト厳定!! SUPER-HDスペシャルセット〜今がチャンス!! 電話で値切ろう(送料無料)

●ザ・ワークステーションと呼ぶにふさわしい
SUPER-HD=オクト厳定セット!!

※マウス・トラックボール付!!
ディスプレイにはスピーカ2個
チルト台付!!

Ⓐ:CZ-623C-TN+CZ-606D-TN…………定価合計¥577,800▶大特価

12回	¥37,900	24回	¥20,100	36回	¥14,000	48回	¥11,000	60回	¥9,200

Ⓑ:CZ-623C-TN+CZ-613D-TN…………定価合計¥633,000▶大特価

12回	¥41,700	24回	¥22,100	36回	¥15,400	48回	¥12,100	60回	¥10,200

現金特価!!
お電話下さい。

上記セットお買い上げの方に、オクトからのプレゼント!!
①MD-2HD 10枚
②シムシティ(ゲームソフト¥8,800)
③ジョイカード(連射式)
④シリコンキーボードカバー(¥2,800)

※超低金利クレジットをご利用下さい。1回〜60回払い、頭金ナシ!! ボーナス1回及び2回払いOKです。

X68000ソフト大セール実施中※ゲームソフトオール25%off 送料¥500

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0
(シャフト)定価¥58,000…………特価¥39,400

〈データーベース〉●KAMIKAZE
定価¥68,000…………特価¥45,400

〈グラフィック〉●C-TRACE68
(キャスト)定価¥68,000…………特価¥51,000

〈C言語〉●C & Professional Pack
定価¥58,000…………特価¥43,000

〈グラフィック〉●サイクロ エキスプレスα68
定価¥97,000…………特価¥73,000

〈グラフィック〉●デジタルクラフト
定価¥39,800…………特価¥28,000

〈ワープロ〉●ハイパーワード
定価¥39,800 CZ-251BS…………特価¥29,800

〈開発ツール〉●C-コンパラPRO68KV.2
定価¥44,800 CZ-245IS…………特価¥33,300

〈CGツール〉●CANVAS PRO68K
定価¥29,800 CZ-249GS…………特価¥22,200

型名	商品	定価	特価
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48.000
CZ-213MS	MUSIC PRO68K	¥18,800	¥13.500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11.500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12.800
CZ-219SS	OS-9/X68000	¥29,800	¥21.000
CZ-220BS	DATA PRO-68K	¥58,000	¥41.000
CZ-223CS	Communication PRO-58K	¥19,800	¥14.300
CZ-224LS	THE 福袋 V2.0	¥9,900	¥7.500
CZ-226BS	CARD PRO-68K	¥29,800	¥21.300
CZ-241BS	システム手帳リフィル集	¥9,800	¥7.500
CZ-242BS	活用フォーム集	¥9,800	¥7.500
CZ-244SS	Honan 68K Ver2.0	¥9,800	¥7.500
CZ-247MS	MUSIC PRO-68K(MIDI)	¥28,800	¥20.800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11.500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15.200
EW		¥38,000	¥29.800
G-68K		¥14,800	¥11.400
E-68		¥19,800	¥15.300
CZ-255GS	CANVASドローグラフィックLIB	¥8,800	¥6.600
CZ-256GS	CANVASドローグラフィックVOL.2	¥8,800	¥6.600
CZ-260LS	XBAS to CHECKER PRO68K	¥9,800	¥7.500
CZ-259SS	SX-WINDOW Ver.1.0	¥6,800	¥5.000
CZ-234LS	AI-68K	¥188,000	¥139.000
CZ-252MS	MUSIC STUDIO PRO68K	¥28,800	¥21.500

モデムコーナー (送料¥1,000)

オムロン
●MD-1200AIII…………特価¥14,500
●MD-12FS…………特価¥15,000
●MD-24FP4II…………特価¥26,000
●MD-24FN5…………特価¥30,000
●MD-24FS4…………特価¥31,000
●MD-24FS5…………特価¥32,000
●MD-24FS7…………特価¥44,000
●MD-24FC5…………特価¥34,000
●MD-24FP5II…………特価¥28,500
●MD-24FN4…………特価¥27,000
●MD-24FJ4…………特価¥31,000
●MD-24FJ5…………特価¥34,000
●MD-24HS…………特価¥64,000
●MD-48HS…………特価¥98,000
●MD-96FS5…………特価¥131,000

AIWA
●PV-A24VM5…………特価¥30,000
●PV-M24VM5…………特価¥30,000
●PV-A12…………特価¥14,500
●PV-M24…………特価¥28,500

熱転写カラー漢字プリンター (ケーブル用紙付) 送料¥1,000

■CZ-8PC5 NEW
NOW PRINTING!!
●48ドット
●熱転写カラー漢字プリンター
定価¥96,800
大特価TEL下さい!!

①CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……大特価!! TEL下さい。
②CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000……大特価!! TEL下さい。
③CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000……大特価!! TEL下さい。
④IO-735×(カラーイメージジェット)
定価¥248,000……大特価¥183,000

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥15,500

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥11,500

周辺機器コーナー (送料¥500)

●CZ-6BE1 IBM増設RAMボード…………(¥35,000)▶特価¥26,500
●CZ-6BE1B IMB増設RAMボード…………(¥28,000)▶特価¥21,000
●CZ-6BE2 2MB増設RAMボード…………(¥79,800)▶特価¥60,500
●CZ-6BE4 4MB増設RAMボード…………(¥138,000)▶特価¥104,800
●CZ-6BF1 増設用RS-232Cボード…………(¥49,800)▶特価¥38,500
●CZ-6BG1 GP-IBボード…………(¥59,800)▶特価¥45,000
●CZ-6BM1 MIDIボード…………(¥26,800)▶特価¥20,500
●CZ-6BN1 スキャナ用パラレルボード…………(¥29,800)▶特価¥22,800
●CZ-6BP1 数値演算プロセッサボード…………(¥79,800)▶特価¥60,500
●CZ-6BO1 ユニバーサルI/Oボード…………(¥39,800)▶特価¥30,500
●CZ-6EB1/BK 拡張I/Oボックス…………(¥88,000)▶特価¥65,800
●CZ-6VT1/BK カラーイメージ・ユニット…………(¥69,800)▶特価¥53,000
●CZ-8NM2A マウス…………(¥68,800)▶特価¥5,300
●CZ-8NT1 マウストラックボール…………(¥98,800)▶特価¥7,500
●CZ-8NS1 カラーイメージスキャナ…………(¥188,000)▶特価¥137,000
●CZ-6BC1 FAXボード…………(¥79,800)▶特価¥60,500
●CZ-8TM2 モデムユニット…………(¥49,800)▶特価¥38,000
●CZ-64H 増設ハードディスク…………(¥120,000)▶大特価
●CZ-6TU GY/BK RGBシステムチューナー…………(¥33,100)▶特価¥25,000
●BF-68PRO 高性能CRTフィルター…………(¥19,800)▶特価¥15,500
●CZ-6MO1 光磁気ディスクユニット…………(¥450,000)▶特価¥333,000
●CZ-6BS1 SCSIインターフェースボード…………(¥29,800)▶特価¥22,200
●CZ-6BL2 LANボード…………(¥298,000)▶大特価

特選周辺機器(送料¥1,000)

●SX-68M MIDインターフェースボード
(システムサコム)¥19,800…………特価¥15,000
●CZ-6BV1 ビデオボード
¥21,000…………特価¥15,500

■増設RAMボード=I・Oデータ
①PIO-6BE1-A(1MB)
¥25,000…………特価¥17,800
②PIO-6BE2-2M(2MB)
¥50,000…………特価¥35,800
③PIO-6BE4-4M(4MB)
¥88,000…………特価¥62,500

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

★通信販売お申込みのご案内★

〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

3回	3.5%	6回	4.5%	10回	6%
12回	6%	18回	11%	24回	12.5%
36回	17.5%	48回	23%	60回	29.5%

振込先
富士銀行 久ヶ原支店 当No.1824
三菱銀行 蒲田支店 当No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

超低金利クレジットC

またまた秋葉原で おなじみ

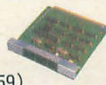
注目!!
(平成3年3月末はもちろん)
平成3年4月末一括払い
手数料(金利)無料
ご利用下さい。

2/15～3/14

●お近くの方は
●本体単品で
●ビジネスソフト

HARD DISK UNIT(X68000専用)
アイテック(SCSI)
●ITX-80S(80MB/20ms)…定価¥128,000▶特価¥ 95,000
●ITX-130S(130MB/20ms)…定価¥158,000▶特価¥117,000

Fine Scanner-X68
(HAL研究所)X68000専用
■HGS-68 (定価¥39,800)
特価¥26,500
(送料・消費税込み¥27,810)

X68000シリーズ専用
MIDIインターフェースボード
SX-68M(サコム)
特価¥14,800
(純生コンパチ)定価¥19,800
(送料・消費税込み¥15,759)

X68000メモリボード(シャープ&I/O・DATA)(送料¥500)

①CZ-6BE1(600C用)
定価¥35,000……¥26,000
(送料・消費税込¥27,295)
②PIO-6BE1-A
定価¥25,000……¥18,000
(送料・消費税込¥19,055)
③PIO-6BE2-2M
定価¥50,000……¥35,000
(送料・消費税込¥36,565)
④PIO-6BE4-4M
定価¥88,000……¥62,000
(送料・消費税込¥64,375)

ジョイスティック 送料¥500
●X-1PRO
定価¥9,500▶特価¥7,80
●ASCII STICK
定価¥6,800▶特価¥5,50

P&A超低金利クレジットをご利用ください!!

NEW X68000 SUPER/SUPER-HD/PROII/PROII-HD (送料・消費税込)

SUPER
セットでお買い上げの方に●ディスケット10枚 ●ジョイカード2ケプレゼント中!!

NEW

Ⓐセット:CZ-604C-TN+CZ-606D-TN……定価¥427,800▶特価 価格はTEL下さい。

12回	27,000	24回	14,300	36回	9,900	48回	7,800	60回	6,600

Ⓑセット:CZ-604C-TN+CZ-613D-TN……定価¥483,000▶特価 価格はTEL下さい。

12回	30,400	24回	16,100	36回	11,200	48回	8,800	60回	7,400

SUPER-HD
セットでお買い上げの方に●ディスケット10枚 ●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-623C-TN+CZ-606D-TN……定価¥577,800▶特価 価格はTEL下さい。

12回	36,300	24回	19,300	36回	13,400	48回	10,500	60回	8,900

Ⓑセット:CZ-623C-TN+CZ-613D-TN……定価¥633,000▶特価 価格はTEL下さい。

12回	39,700	24回	21,000	36回	14,600	48回	11,500	60回	9,700

PROII
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-653C+CZ-606D……定価¥364,800▶特価 価格はTEL下さい。

12回	21,800	24回	11,500	36回	8,000	48回	6,300	60回	5,300

Ⓑセット:CZ-653C+CZ-605D……定価¥400,000▶特価 価格はTEL下さい。

12回	24,200	24回	12,800	36回	8,900	48回	7,000	60回	5,900

Ⓒセット:CZ-653C+CZ-604D……定価¥798,000▶特価 価格はTEL下さい。
Ⓓセット:CZ-653C+CZ-613D……定価¥420,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-653C+CU-21HD……定価¥433,000▶特価 価格はTEL下さい。

PROII-HD
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-663C+CZ-606D……定価¥474,800▶特価 価格はTEL下さい。

12回	30,200	24回	16,000	36回	11,200	48回	8,800	60回	7,400

Ⓑセット:CZ-663C+CZ-605D……定価¥510,000▶特価 価格はTEL下さい。

12回	32,300	24回	17,100	36回	11,900	48回	9,300	60回	7,900

Ⓒセット:CZ-663C+CZ-604D……定価¥489,800▶特価 価格はTEL下さい。
Ⓓセット:CZ-663C+CZ-613D……定価¥530,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-663C+CU-21HD……定価¥543,000▶特価 価格はTEL下さい。

セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

◎電話にて、ドンドンお問合せ下さい!!
クレジット表には、出せないほどの価格です。
メーカーさん、ご免なさい。
ユーザーの方には大歓迎されそうです。
今がチャンスです、ハイ。

X68000シリーズ ～P&Aスペシャルセット=限定誌上販売!!

台数限定 送料、消費税込み ※セットでお買い上げの方に、●ディスケット10枚、●ジョイカード2個 プレゼント中!!

EXPERTII

Ⓐセット:P&A厳選セット
■CZ-603C
(本体価格¥338,000)
+
■CZ-606D
(モニター定価¥79,800)
▶P&A超特価 ¥304,000

Ⓑセット
■CZ-603C+CZ-604D
定価¥432,800…▶特価¥309,000
Ⓒセット:
■CZ-603C+DZ-605D
定価¥453,000…▶特価¥322,000
Ⓓセット:
■CZ-603C+CZ-613D
定価¥473,000…▶特価¥342,000
Ⓔセット:
■CZ-603C+CU-21HD
定価¥486,000…▶特価¥347,000

EXPERT-HD

Ⓐセット:P&A厳選セット
■CZ-612C(ブラック)
(本体価格¥466,000)
+
■CZ-606D(ブラック)
(モニター定価¥79,800)
▶P&A超特価 ¥335,000

Ⓑセット:
■CZ-612C+CZ-604D
定価¥560,800…▶特価¥340,000
Ⓒセット:
■CZ-612C+CZ-605D
定価¥581,000…▶超特価¥359,000
Ⓓセット:
■CZ-612C+CZ-613D
定価¥601,000…▶超特価¥372,000
Ⓔセット:
■CZ-612C+CU-21HD
定価¥614,000…▶超特価¥386,000

■NEC=モデム(定価¥44,800)
⦿COMSTARZ 2424/5
●2400/1200bps全二重
●MNP5クラス
●インターフェース付
P&A超特価
¥27,500
(送料・消費税込み¥29,355)

■ALL in Note
フリートップ
パーソナルコンピュータ
⦿AX-286N-H2
(定価¥398,000)
P&A超特価
価格はTEL下さい。

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00～PM7:00、日祭AM10:00～PM6:00

~84回払いまでOK!!

★頭金なし!★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

寄り下さい。専門係員が説明いたします。
で受付します。詳しくは電話にてお問合せ下さい。
の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー（送料1ヶ～5ヶまで￥500）

商品	定価	特価
Z's STAFF PRO68K Ver.2.0（ツァイト）	¥ 58,000	¥ 39,500
Z's TRIPHONY デジタルクラフト（ツァイト）	¥ 39,800	¥ 27,800
テラッツォ（ハミングバード）	¥ 19,400	¥ 14,200
KAMIKAZE（サムシング・グッド）	¥ 68,000	¥ 44,800
C & Professional Pack（マイクロウェアジャパン）	¥ 58,000	¥ 43,400
Final Ver3.2（エーエスピー）	¥ 38,000	¥ 29,600
C-compiler PRO68K Ver.2 CZ-245L	¥ 44,800	¥ 33,300
CARD PRO68K CZ226BS	¥ 29,800	¥ 21,200
XBAS to C CHECKER CZ-260LS	¥ 9,800	¥ 7,400
OS-9/X68000 CZ219SS	¥ 29,800	¥ 22,500
AI-68K CZ234LS	¥ 188,000	¥138,000
THE 福袋 V2.0 CZ224LS	¥ 9,900	¥ 7,400
SOUND PRO68K CZ-214MS	¥ 15,800	¥ 11,400
MUSIC PRO68K CZ213MS	¥ 18,800	¥ 13,400
Sampling PRO68K CD215MS	¥ 17,800	¥ 12,700
MUSIC-studio PRO68K CZ-252MS	¥ 15,800	¥ 21,400
MUSIC-PRO68K〔MIDI〕247MS	¥ 28,800	¥ 20,700
New-print Shop 221HS	¥ 19,800	¥ 15,500
Communication 223CS	¥ 19,800	¥ 14,200
Communication Ver.2 CZ-257CS	¥ 19,800	¥ 15,500
C-TRACE68 Ver.3.0（キャスト）	¥ 98,000	¥ 74,600
サイクロンEXPRESS α68	¥ 98,000	¥ 69,800
G68K Ver2 PRO	¥ 22,000	¥ 17,500
SX-WINDOW CZ-259SS	¥ 6,800	¥ 4,900
Gツール（ザインソフト）	¥ 28,000	¥ 18,900
たーみのる2（SPS）	¥ 17,800	¥ 13,300
マジックパレット（ミュージカルプラン）	¥ 19,800	¥ 14,500
Hyper word CZ-251BS	¥ 39,800	¥ 29,600

●ゲームソフト20%OFF OK!!（一部ソフト除く）

周辺機器コーナー（送料￥500）

	商品	定価	特価
Ⓐ	CZ-8NSI	￥188,000	￥145,000
Ⓑ	CZ-6VTI	￥ 69,800	￥ 52,500
Ⓒ	CZ-6TU	￥ 33,100	￥ 24,500
Ⓓ	BF-68PRO	￥ 19,800	￥ 15,300
Ⓔ	CZ-6BEI	￥ 35,000	￥ 26,000
Ⓕ	CZ-6BEIA	￥ 38,000	￥ 28,600
Ⓖ	CZ-6BE2	￥ 79,800	￥ 60,000
Ⓗ	CZ-6BE4	￥138,000	￥103,000
Ⓘ	CZ-6BFI	￥ 49,800	￥ 38,200
Ⓙ	CZ-6BPI	￥ 79,800	￥ 60,000
Ⓚ	CZ-6BMI	￥ 26,800	￥ 20,300
Ⓛ	CZ-6EBI	￥ 88,000	￥ 66,500
Ⓜ	AN-S100	￥ 36,600	￥ 28,500
Ⓝ	CZ-6SDI	￥ 44,800	￥ 35,000
Ⓞ	CZ-6BN1	￥ 29,800	￥ 22,600
Ⓟ	CZ-6BV1	￥ 21,000	￥ 15,900
Ⓠ	CZ-64H	￥120,000	￥ 91,500
Ⓡ	CZ-6BG1	￥ 59,800	￥ 45,000
Ⓢ	CZ-6BU1	￥ 39,800	￥ 30,300
Ⓣ	CZ-6PVI	￥198,000	￥153,000
Ⓤ	IO-735X	￥248,000	￥190,000
Ⓥ	CZ-6BS1	￥ 29,800	￥ 22,300
Ⓦ	CZ-8NJ2	￥ 23,800	￥ 18,500
Ⓧ	CZ-6BL2	￥298,000	￥222,000

X68000用ハードディスク（送料￥1,000）

アイテム

商品	定価	特価
HXD-040（40MB/23ms）	￥118,000	￥ 88,000
HXD-042（増設用）	￥128,000	￥ 95,000

アイテック

商品	定価	特価
ITX-640（40MB/28ms）	￥158,000	￥ 83,000
ITX-680（80MB/20ms）	￥198,000	￥ 97,000

プリンター（ケーブル・用紙付）（送料￥1,000）

商品	定価	特価
CZ-8PC5-BK NEW	￥ 96,800	￥72,500
CZ-8PK10	￥ 97,800	￥73,000
CZ-8PG2	￥160,000	価格はTEL!!
CZ-8PG1	￥130,000	価格はTEL!!

モデムコーナー（送料￥1,000）

	商品	定価	特価
Ⓐ	MD-24FS5（オムロン）	¥ 49,800	¥31,500
Ⓑ	MD-24FS7（オムロン）	¥ 64,800	¥43,500
Ⓒ	コムスター2424/4（NEC）	¥ 38,800	¥25,900
Ⓓ	コムスター2424/5（NEC）	¥ 44,800	¥27,500

P & A 特選パソコンラック（送料無料）移動自由（キャスター付）

中古パソコン（セットはモニター付）送料￥2,000

商品	価格
X68000セット	￥180,000
X68000 ACEセット	￥200,000
X68000 ACE-HDセット	￥215,000
EXPERTセット	￥230,000
EXPERT-HDセット	￥265,000
PROセット	￥250,000
X68000PRO-HDセット	￥270,000
EXPERT IIセット	￥250,000
EXPERT II-HDセット	￥320,000
PRO IIセット	￥240,000
PRO II-HDセット	￥310,000

超特価でクレジットが組める!!

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合…………価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）

●買取りの場合…………現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々￥1,000円からOK!! ●ボーナス払いOK（夏冬10回までOK）
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.5	4.5	6.0	6.0	11.0	12.5	17.5	23.0	29.5	38.0	45.5

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます（祭日の場合は翌日になります）

●マイコン
●ビデオ
●ビデオテープ

P&A

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148（代）
FAX.03-3651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-3770-8855

■アフターサービス万全のサポート体制
- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…。午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

■TEL・FAXのお見積OK.//
■低金利クレジットをご利用下さい。

▶2・15〜3・14

SHARPのことなら なんでもおまかせ.//

大徳買セール/ 安く値切ってネ。
お電話下さい。㊙価格をお知らせいたします。

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2
(定価 ¥23,800)
OAランド特価 ▶¥18,000

電子手帳 ●見やすい漢字4桁表示.// 情報任時代の必需品.//
- ■PA-9500(¥48,000)…▶特価¥38,000
- ■PA-8500(¥28,000)…▶特価¥15,000
- ■PA-7500(¥22,000)…▶特価¥12,000

2月中旬より、お店の場所が移転しました。装いも新たなオーエーランドです。ぜひお立寄り下さい。

SHARP X68000シリーズセット どんどんTEL下さい。

X68000 SUPER NEW

①CZ-604C-TN+CZ-613D-TN
定価合計¥483,000

1回	345,000	12回	31,600
24回	16,800	36回	11,700

②CZ-604C-TN+CZ-606D-TN
定価合計¥427,800

1回	306,000	12回	28,000
24回	14,800	36回	10,300

■CZ-604C 特価~~¥348,000~~
■CZ-623C 特価~~¥498,000~~

X68000 SUPER-HD

①CZ-623C-TN+CZ-613D-TN
定価合計¥633,000

1回	456,000	12回	41,500
24回	22,000	36回	15,300

②CZ-623C-TN+CZ-606D-TN
定価合計¥577,800

1回	416,000	12回	37,800
24回	20,000	36回	13,900

X68000 PROII

①CZ-653C+CZ-613D
定価合計¥420,000

1回	280,000	12回	25,400
24回	13,500	36回	9,400

②CZ-653C+CZ-605D
定価合計¥400,000

1回	265,000	12回	24,100
24回	12,800	36回	8,900

③CZ-653C+CZ-606D
定価合計¥364,800

1回	240,000	12回	21,800
24回	11,500	36回	8,000

■CZ-653C 特価~~¥285,000~~
■CZ-663C 特価~~¥395,000~~

X68000 PROII-HD

①CZ-663C+CZ-613D
定価合計¥530,000

1回	383,000	12回	35,000
24回	18,000	36回	12,900

②CZ-663C+CZ-605D
定価合計¥510,000

1回	370,000	12回	33,600
24回	17,800	36回	12,400

③CZ-663C+CZ-606D
定価合計¥474,800

1回	344,000	12回	31,300
24回	166,000	36回	11,500

上記組合せのディスプレイ(モニター)変更自由.//
詳しくは、お電話にてお問い合せ下さい.//

■期間中、セットでお買い上げの方には、①Vボール②ニュージーランド・ストーリー(ゲーム)のがついてきます。さらに、③テトリスやドルアーガの塔などの入ったゲームパックもプレゼント.//

X68000用SCSIハードディスク.//

キャラベル
- ①AV-040SC+ケーブル……特価¥79,000 (合計定価¥116,000)
- ②AV-090SC+ケーブル……特価¥114,000 (合計定価¥172,000)
- ③AV-130SC+ケーブル……特価¥139,000 (合計定価¥212,000)
- ④AV-200SC+ケーブル……特価¥208,000 (合計定価¥314,000)
- ⑤AV-250SC+ケーブル……特価¥266,000 (合計定価¥402,000)

アイテック
- ①ITX-80S……特価¥88,500 (定価¥128,000)
- ②ITX-120S……特価¥108,000 (定価¥158,000)

※X68000 SUPER 以外の機種では、SCSIボードが必要となります。
⊙SCSIボード……特価¥22,000

周辺機器

■光磁気ディスクユニット
●CZ-6MO1 (定価¥450,000)
特価TEL下さい.//

■SCSIボード
●CZ-6BS1 (定価¥29,800)
特価¥22,300

■ビデオボード
●CZ-6BV-1 (定価¥21,000)
特価¥15,600

周辺機器コーナー 電話で値切ろう。

プリンターセットコーナー

①CZ-8PC5 NEW 定価¥96,800
●48ドット●熱転写カラー漢字プリンター
大特価TEL下さい.//

②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……特価¥71,500

③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥93,500

④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥114,800

OAランド特選品.//

■IO-735X(定価¥248,000)
●カラーイメージジェットプリンター ●ケーブル付
特価¥183,000

モデム

オムロン	MD-1200A III	¥14,500
	MD-24FP4 II	¥27,500
	MD-24FP5 II	¥29,800
	MD-24FN4	¥28,000
	MD-24FN5	¥31,300
	MD-24FJ4	¥31,300
	MD-24FJ5	¥34,500
	MD-24FS4	¥28,500
	MD-24FS5	¥34,500
アイワ	PV-A24VM5	¥32,500
	PV-M24	¥28,800
NEC	COMSTAR 2424/4	¥28,800
	COMSTAR 2424/5	¥33,500

X68000用周辺機器コーナー

- ①CZ-6VT1(カラーイメージユニット) 定価¥69,800……特価¥52,500
- ②CZ-8NS1(カラーイメージスキャナー) 定価¥88,000……特価¥138,000
- ③CZ-6BM1(MIDIボード) 定価¥26,800……特価¥20,500
- ④CZ-8NJ2(インテリジェント・コントローラー) 定価¥23,800……特価¥18,000
- ⑤CZ-6TU(RGBシステムチューナー) 定価¥33,100……特価¥25,000
- ⑥CZ-64H(増設ハードディスク) 定価¥120,000……特価¥89,000
- ⑦CZ-6EB1(拡張I/Oボックス=4スロット) 定価¥88,000……特価¥66,000
- ⑧CZ-6BP1(数値演算プロセッサボード) 定価¥79,800……特価¥60,000

I・Oデータ増設RAMボード

■PIO-6BE2-2M (2MB) 定価¥50,000 特価¥35,500
■PIO-6BE1-A (1MB) 定価¥25,000 特価¥17,800
■PIO-6BE4-4M (4MB) 定価¥88,000 特価¥62,200

《計測技研》

●高速増設メモリと数値演算プロセッサが一つのボードになった.//

型番	定価	特価
●KGB-X68PRK-00	(¥34,000)	特価¥26,000
● -01	(¥58,000)	特価¥43,500
● -02	(¥74,000)	特価¥55,500
● -03	(¥98,000)	特価¥73,500
● -04	(¥122,000)	特価¥91,500
● -10	(¥72,000)	特価¥54,000
● -11	(¥96,000)	特価¥72,000
● -12	(¥112,000)	特価¥84,000
● -13	(¥136,000)	特価¥102,000
● -14	(¥160,000)	特価¥120,000

OAランド今月の大目玉.//=超A級中古品

- ●CZ-603C-GY……特価¥200,000
- ●CZ-613D-GY……特価¥79,000
- ●CZ-603C-BK……特価¥218,000 (メーカー保証付)
- ●CZ-8PK8(2台)……特価¥40,000
- ●CZ-8PK9……特価¥38,000
- ●CZ-8NS1……特価¥113,000
- ●CZ-6MO1……特価¥330,000 (メーカー保証付)
- ●CZ-8PC5……特価¥72,000

早い者勝ちですので、ドンドンTELお待ちしてます.//

OAランド推奨ソフト

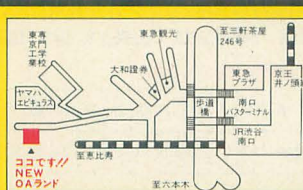
- ■SX-WINDOW (次世代インテリジェントソフト) 定価¥6,800 特価¥5,100
- ■CZ-245LS (C-コンパイラII) 定価¥44,800 特価¥33,500
- ■CZ-260LS (X Bas to C CHECKER) 定価¥9,800 特価¥8,000
- ■CZ-249GS (CANVAS-PRO68K) 定価¥29,800 特価¥22,300
- ■CZ-255GS/256GS (ドローグラフィックライブラリ1/2) 定価¥8,800 特価¥7,000
- ■CZ-219SS (OS9/X68,000) 定価¥29,800 特価¥23,800

通信販売のご案内

―全国通販―

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■年中無休です.//

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

クレジット表

3回	3.5%	6回	4.5%	10回	6%	12回	6%	15回	8.5%	18回	11%	20回	12%
24回	12.5%	30回	17%	36回	17.5%	42回	22.5%	48回	23%	54回	29%	60回	29.5%

㈱オー・エー・ランド

〒150 東京都渋谷区桜丘町3-13 アルカディア2F
☎(03)3770-8855
関東エリアの送料は、1個につき¥1,000です。 FAX(03)3770-7080

★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、1月下現在です。

システムソフトが広げる、面白くする、X68000エキサイティング・シーン。

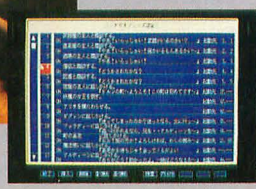
SystemSoft X68000series Line up

BRETONNE LAIS

新発売

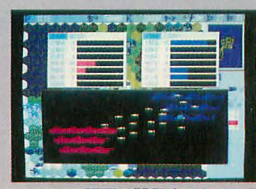
ブルトン・レイ シナリオエディタ

想像から創造へ。君だけの英雄伝説。

神秘的な中世ファンタジーの世界を舞台にした正統派ロールプレイングゲーム——ブルトン・レイ。その物語を自由に創造することができる「シナリオエディタ」が、ついにX68000に登場。町や平原、山を配置し、登場人物を選び、アイテムを設定し、シナリオを組み立て、そして感動のエンディングへ。多彩な機能と使い勝手のよい操作性で、誰もが壮大なストーリーの創造主になれるのだ。しかも、98シリーズとのデータ互換も可能。幾多の物語を、X68000に再び蘇らせることもできる。「シナリオエディタ」は、すべての者をブルトン・レイの世界へさらに深く引きずり込む、必携のクリエイティブ・ツールだ。

■X68000シリーズ　■5″-2HD(2枚組)
- 別売の「ブルトン・レイ」X68000版が必要です。
- アナログRGB(31KHz対応)ディスプレイをお使いください。
- メインメモリが2MB以上の場合、日本語入力フロントプロセッサとしてASK68Kが使用できます。(2MB未満の場合は、単漢字入力となります。)

価格 5,800円

戦略は、いま新たなる次元へ。

つねに未知なる戦略シーンを追い求める勇者たちへ、新たな興奮と感動を贈ろう。ストラテジック・シミュレーションの頂点に立つ不朽の名作「大戦略シリーズ」が、ついにX68000に登場。「キャンペーン版大戦略II」—通常のマップモードに加え、戦闘経験をへて熟練度の上がったユニットをそのまま引き継いで次のステージに進み、8ステージの連続制覇に挑む〈キャンペーンモード〉を導入。都市に、孤島に、そして平原に展開する熾烈な戦いの舞台を、じっくり楽しむことができる。ウインドウメニュー、マウス対応など操作性も一段とアップ。いまX68000の広大なフィールドで、戦略は確かな進化を迎えるだろう。

■X68000シリーズ　■5″-2HD(2枚組)
- アナログRGB(31KHz対応)ディスプレイをお使いください。

価格 9,800円

※画面は開発中のものです。

発売日等の最新情報を下記のとおりテレフォンサービスにてご案内いたしております。どうぞお気軽にご利用ください。

新製品の発売日および内容のご案内は……
テレフォンサービス専用電話　東京：03-3326-8710
福岡：092-752-2602

商品のお申し込みおよび発売日に関するお問い合わせは……
営業部専用電話 092-752-5262
土曜日、日曜日、祝祭日は営業いたしておりません。

商品に関する技術的なお問い合わせは……
ユーザーサポート専用電話 092-752-5278
月～金 9:00～12:00　13:00～17:00(祝祭日を除く)

■総合カタログをご希望の方は請求券をはがきに貼り、住所・氏名・年齢・電話番号・使用機種名を明記の上、弊社宛にご送付ください。

※製品の仕様は、機能・性能の改善のため将来予告なしに変更することがあります。
※表示価格に消費税は含まれておりません。

株式会社 システムソフト
〒810 福岡市中央区天神3丁目10-30

総合カタログVol.8請求券
Oh! X 3月号
有効期限：1991年3月末

SOFT BANK

「力作コンテスト」の入選発表は24日、凄いパーティになりそうだ。

Rhodes BOSS D.T.M.S Roland

ROLAND SOUND PARTY

音と音楽のスーパー・イベント

もうすぐです、もうすぐ凄いパーティが爆発します。ローランドの凄い新製品たちの
オンパレード・クリニックや面白く役に立つセミナー、豪華なプレゼント抽選。
そして土橋安騎夫(Rebecca)&本田恭之(Grass Valley)、クライズラー&カンパニーと、今をときめく
強力パフォーマーたちを迎えたスペシャル・ライブなどなど
熱気と興奮の2日間。もちろん新製品をあなた自身が弾いて試すのは一日中フリー・タイム。
さらに3月24日はDTMファンお待ちかね「第3回・力作コンテスト」の受賞発表、
「ドラゴンクエスト」でおなじみのすぎやまこういち氏の特別講演もあって。
先着100名にはスペシャル・プレゼントもうれしくて。では、次にお会いするのは会場で!

平成3年**3月23日(土)/24日(日)** 会場:アート・フォーラム六本木

23(土) 開場:AM11:00~PM8:00

メインステージのイベント・スケジュール	
00~	話題の新製品 オンパレード・クリニック
45~	RhodesとBOSSの劇的インプレッシブ・ライブ
15~	ローランド・リズム・パーティ ハウス・ミュージックにのせて一挙公開 ローランド・リズムマシン・ヒストリー
30~	スペシャル・ライブ 土橋安騎夫(Rebecca)&本田恭之(Grass Valley) スペシャル・ユニット・ライブ! ●ゲストあり(予定)

3/24(日) 開場:AM10:00~PM6:00

メインステージのイベント・スケジュール	
11:15~	話題の新製品 オンパレード・クリニック
12:30~	「力作コンテスト」受賞作品発表
13:30~	「DTMに、いかにハートをこめるか?」講演:すぎやまこういち
14:00~	DTMサウンド・プロデュース講座 by 篠田元一
16:45~	スペシャル・ライブ クライズラー&カンパニー

★★入場には入場整理券が必要です。東京および東京近郊の楽器店/コンピュータ・ショップで配布中。
★★スペシャル・ライブは満員の際、入場できない場合があります。そこで優待チケットが安心。抽選で各200名にさしあげます。お申込み方法は店頭のチラシをご覧ください。
★★サウンド・パーティのお問合わせはローランドMCクラブ☎03-3251-2833へ。

主催 Roland ROLAND MC CLUB ■協賛 シャープ株式会社 ソフトバンク株式会社 NEC 日本電気株式会社 富士通株式会社 ■後援 ソフトメーカー各社 ■協力 全国有名楽器店/パソコン・ショップ

〈資料請求番号S08〉

DōGA・CGアニメーション講座

今月は，間近にせまったCGコンテストの一次審査通過者の作品を中心にお届けします。

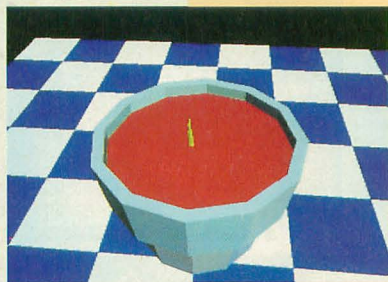
が，まずはその前に“今月のアップデータ”。「花」です。植木鉢のちっちゃな芽が，すくすくと育ってつぼみをつけ，きれいな花を咲かせるまでを見事に表現した作品です。作者の森山さんはCGコンテストのほうでも一次審査通過を果たしています。

さて，CGコンテストの作品です。

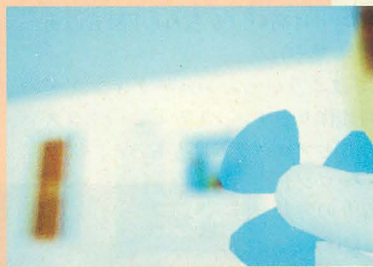
まず，KMC（京大マイコンクラブ）の「CLOCK」。遠近感や視点などがよく研究されていて，楽しめる作品です。KMCからは，ほかに「ゲッピーロボ」や「デスペラード」などがエントリーされています。

次は寺尾響子さんの「HEART」。トランプが笛を吹いている様子を，かわいらしく表現しています。ほのぼのしていて，とっても好感の持てる作品です。

最後に森山知巳さんの「SWORD」。１月号で紹介しているので，くわしくはそちらをご覧ください。

さあ，栄冠は誰の手に!?

今月のアップデータ

花
植物の成長を題材にした作品。きれいに花が咲く様子をCGでうまく描いています。

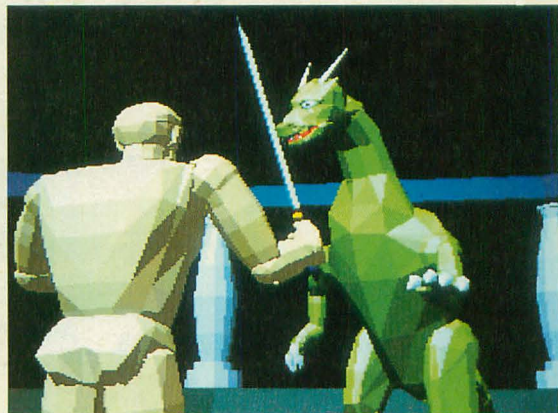
KMC（京大マイコンクラブ）

CLOCK
マッピングや遠近法など，KMCのCGA技術がたっぷり堪能できる作品。初心者の方，必見です。

ゲッピーロボ
おなじみ〝ピー〟シリーズ。厚紙で試作された高さ40cmの模型もあるそう。凝り方が違う！

デスペラード
原作は，KMC内で書かれたSF巨編。テーマ曲，ゲームまでちゃんとあるという作品です。

寺尾 響子さん

HEART
キャラクターがとってもかわいい作品。BGMにはジャズのスタンダードを使用。

森山 知巳さん

SWORD
恐怖の人体モデル作品。これっていったい何時間かかっているのかと考えると……！

今月はX68000用のシューティングゲームと先月予告したX1turboZ専用シューティングゲームの完成直前（？）速報を紹介する。完成にはもうしばらくかかりそうだが,完成次第再度紹介する予定だ。

PARORAN

どこかで見たような自機にどこかで見たような敵が襲いかかってくる。背景は基本的に2重スクロール,スプライト,ガシガシの横スクロールシューティングだ。MIDI対応したBGMや処理速度など,技術力はなかなかのものだ。

画面中から現れる大量のザコキャラ,雪崩のように突進するデカザコの群れ,そしてボスキャラ……,と画面を埋め尽くさんばかりの猛攻が続く。

対する自機側はカプセルでパワーアップ。フォースユニットが強力なので鬼のような攻撃にも耐えられる,という面もあるがフォースをなくしたときでも意外なくらい弾の隙間を抜けられるのだ。単にマシンの表示能力いっぱいに無茶苦茶やっているわけではない。けっこうバランスよくまとまっているようだ（ただし,ウエイトありモード時）。もの足りないというハードシューターにはウエイトなしモードをおすすめする。

ウエイトつきモードでは,あくまで最近のX68000用シューティングゲームと比較してだが,難易度は中程度からやさしい部類となる。敵の攻撃は,物量的には市販ゲームを遥かに上回る。しかし,こちらのパワーアップも強力なので,力押しで抜けられる。いい意味でも悪い意味でも嫌らしい敵キャラは出てこない。

ステージ1は「その道のプロ〈さすらいの花火師編〉」ボスキャラは巨大ヘリコプタのダブルアタックだ。

MIDIによってMT-32系の楽器に対応した音楽もポイントは高い（もちろんFM音源＋AD PCMでも鳴る）。

X68000ではこういったシューティングゲームは比較的作りやすい。しかしこのクラスの作品となるとプログラムはもちろん，グラフィック，サウンドなど，かなりの労力が必要だったはずだ。

入手希望者は無記名の郵便小為替2,000円分と自分の郵便番号住所氏名を明記した宛名シール(2枚必要),および申し込み内容を明記した手紙類を同封のうえ,

〒373　群馬県太田市竜舞892-1

小林様方　Hayabusa Soft

まで封書で問い合わせてほしい。

いよいよステージ3。「X68000Vの瞳」というタイトル。もっとも至烈な面。3連チャンのボスキャラ攻撃に耐えられるかな？

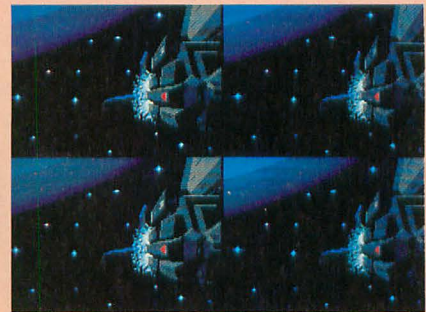
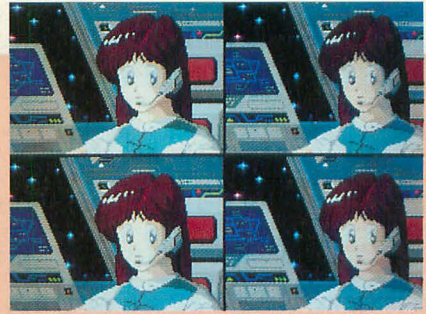
ステージ2は「メカのジェノサイド風味」。もう少しキャラが固ければ，まさに空飛ぶジェノサイド！　反射弾が有効かな？

完成間近Lydion

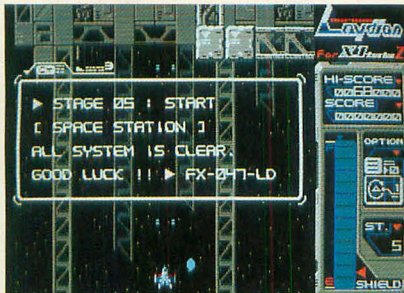
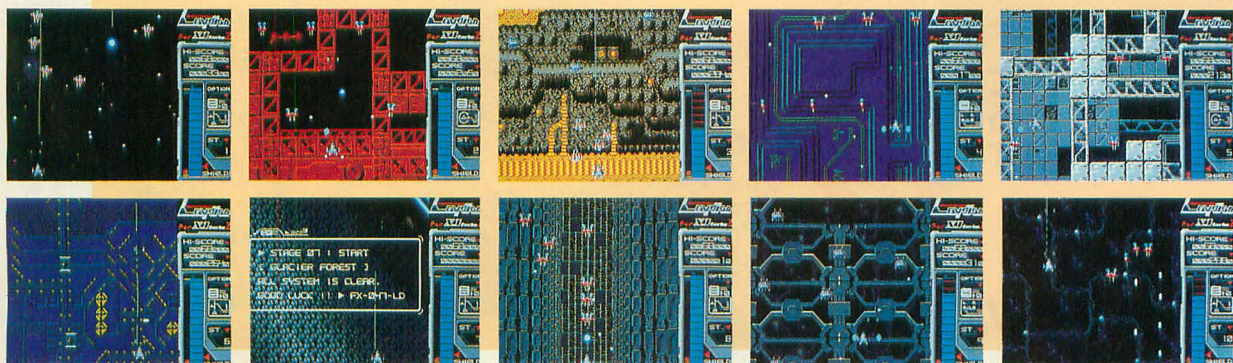
X1turboZ用の……は完成が遅れているため,今回は見送りとさせていただく。といっても,予告した手前なにもなしというのも寂しいので「完成直前! ver.0.96」の画面写真の一部を掲載しておこう。すでにお気づきの方も多いと思うが,以前STUDIO Xで紹介された原秀樹君のX1turboZ専用シューティングゲームだ。

「Cyber Shooter Lydion」というタイトルのあと,ノリのいい音楽とともに堂々たるオープニングデモが始まる。内容はご覧のとおりの縦スクロールシューティングゲーム,最大8重スクロール。8重といっても重ね合わせがあるわけではないので,正確には8段階といったほうがいいかもしれない。横スクロールではよく見かけるが,縦スクロールでこういうパターンは珍しい。

ちなみに2重スクロール部分は完全な「2重」スクロールだ。もちろん,どちらも8ドット単位のガタガタスクロールなどではない。X1turboZで実現されたG-RAMマルチページの同時表示によって表示の負担が軽減され,従来のようにPCGを併用しなくてもよくなったのだ。X1turboZ専用と銘打つだけのことはある。

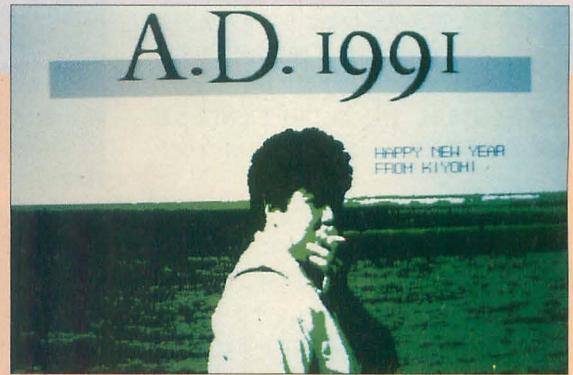
とりあえず面の構成と画面の美しさを示すマップを見ていただきたい。

これがオープニングデモ。4分割アニメーションによるLydion発進シーンが続く。非常に力が入っている。

現時点ではゲームの中心となる敵キャラのパターンが暫定版しか組み込まれてないため,ゲームのスピードや動きの細かさはわかってもゲームバランスはわからない。ボスキャラはある程度できあがっているのだが,それは完成時のお楽しみ。

あとはエンディングとザコキャラパターン自動生成部,ボスキャラの一部,オプション兵器を作って,いろいろ拡張して(最終的には全面2重スクロール対応となる予定らしい),……バグ取りをすれば完成。ま,気長に待ちましょう。

写真はあくまで未完成版。マップの美しさはさすがX1turboZというところ。速度も上々。ちなみに高速スクロールを止めずに撮影すると上のようになる。

1面から10面までのマップを順に並べてみた。それぞれの特長のある面が揃えられている。最終的には2重スクロールになるというが……。

Oh!X readers'ぎゃらりい

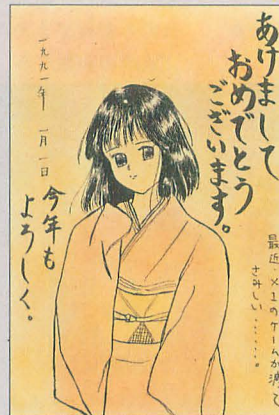
あけましておめでとうのコーナー

新年を迎え，今年もやってきました「あけましておめでとうのコーナー」。皆様，力の入った年賀状をありがとうございました。白黒のはSTUDIO Xのほうで紹介していますのでそちらもどうぞ。

▲笠井清美（北海道）文面より「今年はついに顔が出ました！　女性だと思っていた方，すみません」
◀佐藤昌一（北海道）笠井君が出せっていうもんで

▲丸藤俊之（神奈川県）

▲小井田伸雄（岩手県）

▲平智征（神奈川県）

▲佐藤充浩（長崎県）

▲朝木優子（山口県）

▲福嶋章太（東京都）

いつもお世話さまのスタッフからの年賀状もあるよ

▲福原徹（埼玉県）

▶山田純二（神奈川県）

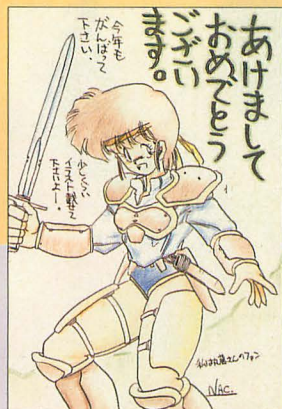
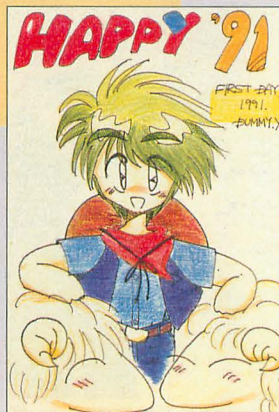
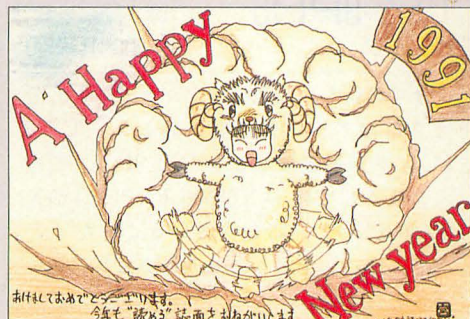
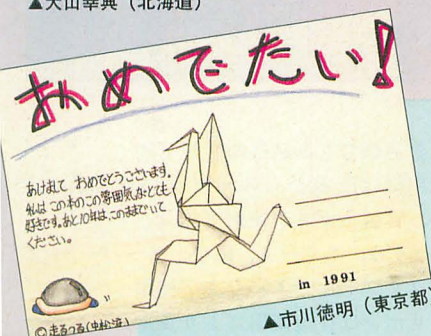
▼筑紫高宏（福岡県）

▼北野雅利（大阪府）

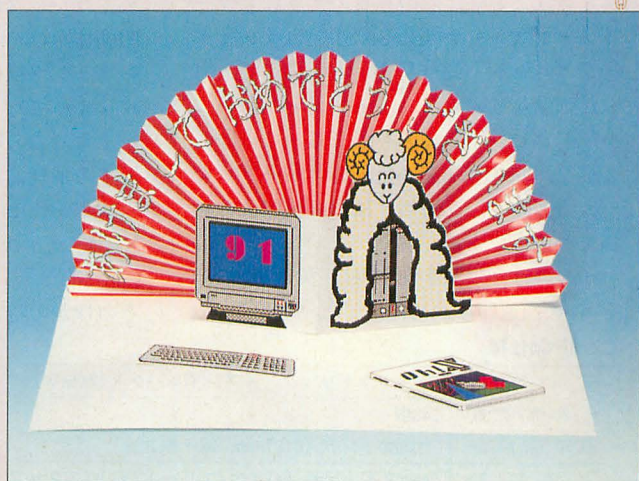
▲板垣修（千葉県）

▲金子聡（新潟県）

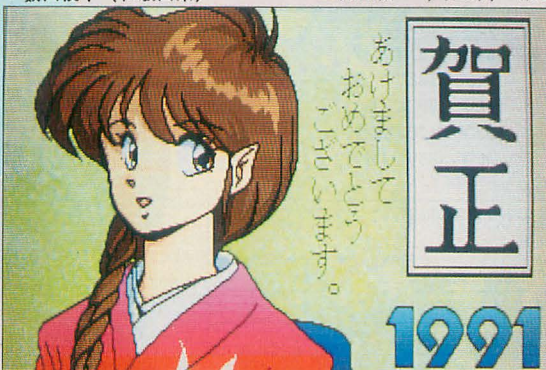
▲佐原功治（愛知県）

▲大山幸典（北海道）

▲板垣央（千葉県）

▲高橋明（茨城県）

▲宮島誠（東京都）

▲市川徳明（東京都）

▲河野純也（宮崎県）

▲泉広明（福島県）

▲横山紘一（埼玉県） 飛び出す年賀状だ！

▲節政暁生（千葉県）

▼藪田俊平（和歌山県）

お知らせ

スタッフを除く23名の方にはOh!X特製記念品を差し上げます。さて，いよいよ5月号は「言わせてくれなくちゃだワ」ということで，皆さんからのいろいろな面白いご意見をお待ちしています。カラーイラストも掲載の予定です。どんどんハガキ出してね。

新製品紹介

周辺機器&ソフトウェア近況レポート

X68000の周辺機器やアプリケーションソフトは，もうかなり充実しているといえるでしょう。というわけで，目新しい製品はありませんが，現在発売中のもののバージョンアップ製品を中心に周辺機器およびソフトの新製品を紹介いたしましょう。

CZ-8PC5によるカラーハードコピー例（縮小率62%）

周辺機器

48ドット熱転写カラー漢字プリンタ

CZ-8PC5

シャープ　96,800円(税別)

まずは新しい周辺機器，シャープから発売されたプリンタを紹介しよう。「CZ-8PC4」の後継機種となる「CZ-8PC5」である。「CZ-8PC4」はX68000 EXPERTやX68000 PROと同時発売ということだったので，実に2年ぶりのモデルチェンジということになる。

本機は48ドット熱転写カラー漢字プリンタということで，右の写真のようにカラーイメージスキャナ「CZ-8NS1」，カラーイメージユニット「CZ-6VT1」との併用で入力した画像などのカラーイメージをきれいにハードコピーすることもできる。この際に必要な，4096色，65536色画面コピープログラム（X68000，X1用）は取扱説明書の付録についている。

印字のほうは漢字86字/秒，英数カナ130字/秒の高速印字が可能。また，明朝体に加えてゴシック体を標準装備している。この明朝体/ゴシック体の切り替えや，設定などのプリンタ操作は見やすく簡単な操作パネルによって行うことができる。

また，この製品の発売にともない新しく以下の消耗品が発売される。

黒リボンカセット（「CZ-8PC5」用）

NEW マルチタイムリボン（「CZ-8PC5/C4」用）

そのほかの特長

・普通紙でも高品質印字が可能

・用紙はB5縦からB4サイズの単票紙のほか，官製ハガキが使用可能

・パイカ，エリート，縮小，スーパー/サブスクリプトなど多彩な文字種類

・リボンは往復使用により長寿命，交換は簡単なリボンカセット式でリボンのみの交換も可能（ランニングコスト低減）

・低静音設計による静かな動作音(47dB)を実現

CZ-8PC5仕様

項目	仕様
印字方式	ドットマトリックス・ノンインパクト(熱転写)
発熱素子数	48個
印字分解能	1/360インチ（0.07mm，縦・横共）
文字種類	ANK文字・記号（パイカ/エリート/縮小/スクリプト文字） 漢字（JIS X0208～1983準拠第1/第2水準）他
印字速度	ANK文字・記号（パイカ）　70字（130字)/秒 ANK文字・記号（エリート）　84字（153字)/秒 ANK文字・記号（縮小）　120字（223字)/秒 漢字　46字（86字)/秒 半角文字　93字（172字)/秒 外字　46字（86字)/秒 ※（　)内は高速印字時
改行間隔	1/6，1/8，n/120インチ改行（選択可能） ※電源投入時は1/6インチ
紙送り方式	フリクション・フィード方式
電源・消費電力	AC 100V，50/60Hz，22W（テスト印字時）
外形寸法・重量	幅410×奥行320×高さ115(mm)・6.7kg

CZ-8PC5

X68000SUPER内蔵型増設用ハードディスク

CZ-68H

シャープ　160,000円(税別)

昨年暮れにX68000の新機種としてX68000 SUPERが発売された。「CZ-68H」はこのX68000 SUPER用の内蔵型，増設用3.5インチハードディスクドライブ(容量80Mバイト）である。

X68000 SUPERはX68000 SUPER-HDから内蔵ハードディスクを取り去っただけというべきものであるので，今回発売の「CZ-68H」を内蔵すればX68000 SUPER-HDと同等品ということになる。

取り付けは側面ぶたをはずし，内部のケーブル，抵抗をはずして5本のネジで行い，コンピュータ側の電源コネクタ，および信号コネクタとつなぐだけなので，ユーザー自身の手で簡単にできる。発売は2月20日の予定。

CZ-68H

CZ-68H仕様

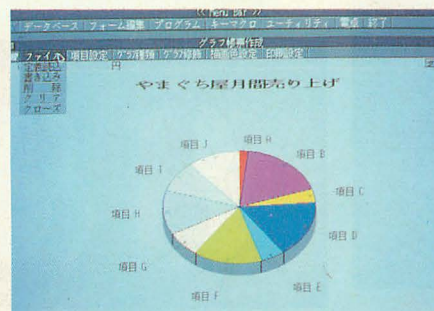
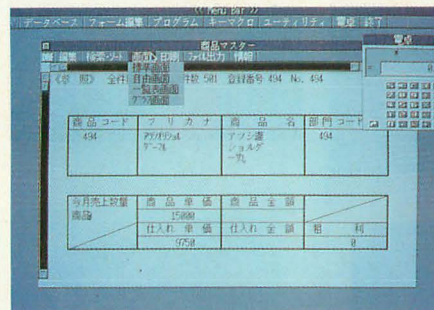
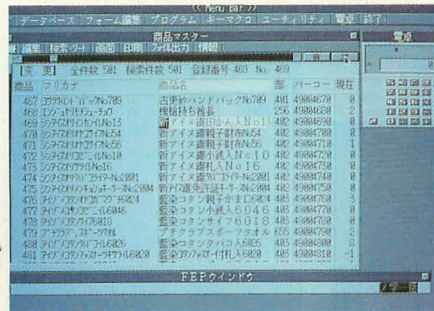
項　目	主な仕様
記憶容量	81Mバイト(フォーマット時)
平均アクセスタイム	19msec
インタフェイス	SCSI規格準拠
使用温度	10〜35℃
消費電力	5W
電源	5VDC，12VDC (コンピュータ本体から受給)
外形寸法	幅101.6×高さ28× 奥行き146(mm)
重量	0.6kg
付属品	取扱説明書，取付ネジ，その他

ソフトウェア

CARD PRO-68K ver.2.0

シャープ　価格未定

㈱ダットジャパンによって開発され，シャープブランドで発売中の「CARD PRO-68K」がバージョンアップ。

新マルチウィンドウシステム搭載（操作性の向上，スピードアップキーボード操作対応)，一覧画面での入力，グラフ機能，キーマクロ機能搭載，ワープロ機能の表面強化，計算機能の強化，子プロセスのサポートなどが主な改良点となっている。3月発売予定。

CARD PRO-68K ver. 2.0

Musicstudio PRO-68K ver.2.0

シャープ　28,800円(税別)

Musicstudio PRO-68K ver. 2.0

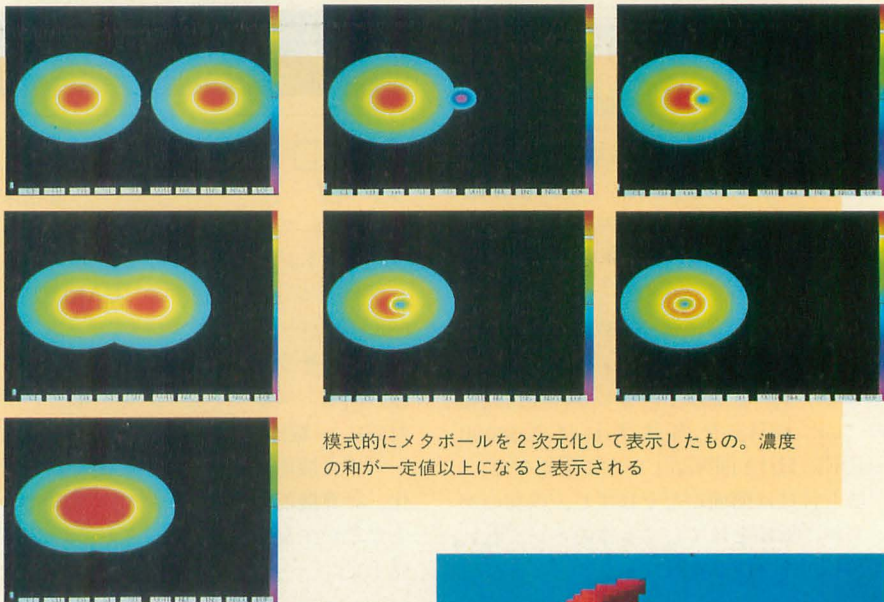
MIDI対応のマルチトラックレコーディングソフト，「Musicstudio PRO-68K」もバージョンアップされた。MIDI，内蔵FM音源，内蔵AD PCM音源の各出力の同期演奏，編集が可能，MUSIC PRO-68K[MIDI]との双方向データコンバート機能，トラック単位，指定範囲（パターン）のデータセーブ機能搭載，というのが主な特長。

「Musicstudio PRO-68K」，および「Musicstudio PRO-68K ver.1.1」のユーザーには有償バージョンアップサービスが行われる。

メタボール対応版
C-TRACE68+
キャスト　198,000円(税別)

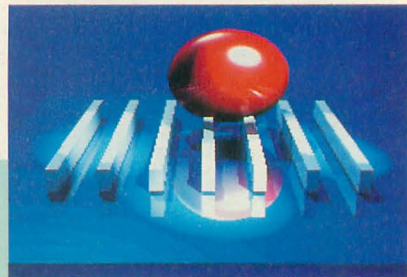
C-TRACE+で新しく加わったメタボール。右はメタボールの考え方を2次元の図として表現してみたもの。右端のカラーバーが濃度変化の目安となる。白い線がしきい値で，物体の表面を表す。1個のメタボールは球体だが2つのメタボールが近づくと融合・変形していくことがわかる。

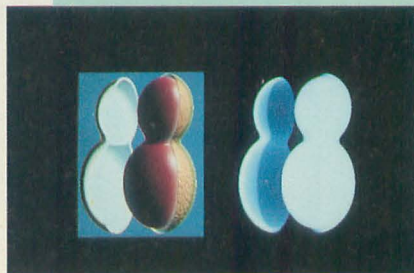
負のメタボールは正のメタボールを削り取っていく。ドーナツ型もこうして2個のメタボールから作られる。

C-TRACE+にはメタボールのほかにもスポット光源やαチャンネルなどの拡張が加えられている。右下はその使用例。αチャンネルではレイトレーシングで計算した物体を任意の画像と自然に（アンチエリアシング，半透明対応）合成できる。

模式的にメタボールを2次元化して表示したもの。濃度の和が一定値以上になると表示される

4，5個のメタボールを使った造形。数値演算プロセッサなしで10時間から20時間かかった（64×64ピクセル）

もっとも基本的な正のメタボール2つがくっついた画像

正負のメタボールで作ったドーナツ型

右はスポット光源の例。光の周辺がぼやけている。下はα合成の例。半透明部に注目

●特集

MIDI & MUSIC PROCESSING

コンピュータミュージックの枠を大きく広げるもの，それがMIDIです。音楽と聞いて逃げ出す人もいるかもしれません。しかし，単に音を鳴らすことがコンピュータミュージックではありません。「楽譜」としてのミュージックデータ，外部機器制御としてのMIDI通信，それらは単に音楽としてだけではなく，パーソナルコンピューティングの世界自体を拡大しうるものです。
特にX68000ではMIDIの普及率も比較的高く，MIDI対応のソフトも増えつつあります。それはMIDIがひとつの必然として我々の前に存在しているからではないでしょうか。未来を展望するとき，システムのなかでMIDIの占める位置は実に明快です。
MIDIの開く世界はある意味で音楽そのものであり，別の意味でより以上の存在ともなります。MIDIの世界をのぞいてみませんか。

CONTENTS

MIDIを使えばこんなに楽しい

コンピュータミュージック入門

Misawa Kazuhiko 三沢 和彦

プロの音楽家にとっては必需品のMIDIですが，なにも楽器の演奏技術がないと使いこなせないというわけではありません。ちょっとした好奇心といくつかの道具さえあれば自分だけのコンピュータ音楽の世界が見えてくることでしょう。

1982年にMIDI(Musical Instrument Digital Interface)規格が現れてから，コンピュータミュージックの発展はめざましいものがあります。いまでは，ロックやニューミュージックの世界でプロの音楽家もコンピュータなしで曲作りすることはほとんどないといってもよいでしょう。プロミュージシャンたちが使っている機材も基本はパソコンとMIDI楽器の組み合わせにしかすぎず，私たちアマチュアパソコンユーザーでも最低限の機材を揃えれば，プロと同じようにコンピュータミュージックを楽しむことができるのです。

そこでこの記事では，ゼロからスタートしたまったくの入門者が自宅でコンピュータミュージックを実現するまでのハウツーをやさしく解説しようと思います。

コンピュータでできること

コンピュータミュージックとは一体どういうものでしょうか。コンピュータを使うのですから，デジタルデータを扱うのだろうというところまでは想像できると思います。ところが，いまや主流のCD(Compact Disk）や最近普及してきたDAT（Digital Audio Tape recorder)もすべてデジタルデータを記録した媒体を使って曲を再生しています。これら一般のデジタル録音とこれから私たちが始めようとするコンピュータミュージックとの違いを知ることが，入門の第一歩です。

まず，CDやDATのしくみをごく簡単に説明しましょう。CDやDATに曲のデータを記録する際にはA/Dコンバータというものを使います（このA/Dコンバータは本誌連載のハードウェア工作入門で詳しく取り上げられました)。これは，曲を構成する音の波形そのものの高低，強弱に対応するデジタルデータに変換してやる装置です。その曲がどの楽器でいくつのパートから成っているかに関わらず，とにかく流れてくる音そのものをそっくりそのままデジタルデータに変換してぺったりと記録してしまうわけです。

次にそのデータを再生するときはD/Aコンバータという逆の装置を使います。これは，記録されているデジタルデータをアナログデータ（すなわち，普通の音）に逆変換してスピーカーから流すのです。これによって，ミュージシャンがスタジオで演奏した曲を私たちが自宅で聞くことができるわけです。

さてそれに対して，コンピュータミュージックでは曲そのもの，すなわち音そのものを記録するのではありません。コンピュータミュージックでは，演奏するための楽譜をデータとして記録するのです。CDに記録された音が，周波数何ヘルツの音が何秒間といった物理的なデータであるのに対して，コンピュータミュージックではどの音階の何分音符といった論理的なデータなのです。

ですから，たとえ実際にミュージシャンがキーボードで直接演奏した曲をそのまま録音したとしても，コンピュータミュージックの形式ではコンピュータによって設定されたテンポにおける，何小節目の何拍目にどのキーがどのくらいの強さで押されたかというデータが記録されるわけです。

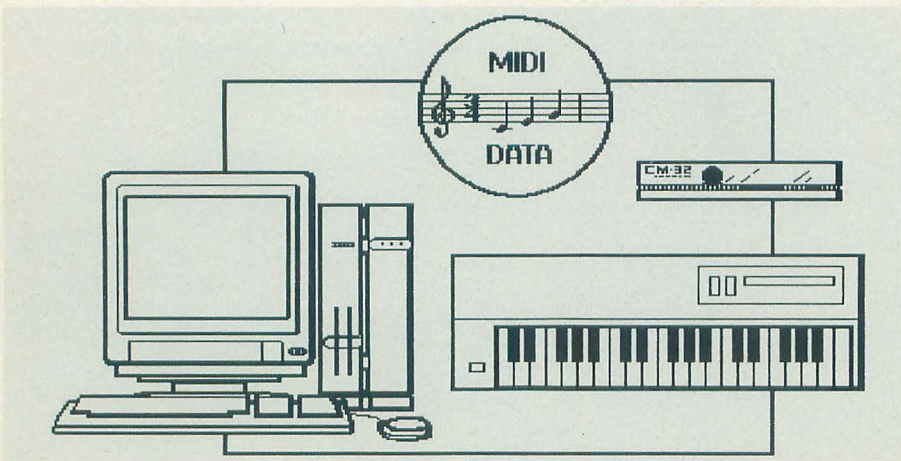
いきなり話が細かくなってしまったかもしれませんが，コンピュータミュージックでできることというのは，楽譜を曲に，あるいは曲を楽譜に変えることとひと言で片づけてもかまわないでしょう。こう考えるとなんでもできそうですね。確かになんでもできるのは間違いありません。実際プロは同じ手順で曲作りをしているのですから。しかし，それだけに難しい面もたくさんあります。そのあたりの詳しいところは，この記事を順番に読み進めていくとわかってくることでしょう。

機材を揃えよう

コンピュータミュージックはなんでもできるということがわかりました。もちろん原理的にという意味で，なにも機材がなければ音すら出ません。そこで，コンピュータミュージックを始めるうえで必要な機材を最低限のところで揃えていきましょう。

1） X68000本体

まず絶対に必要なのはコンピュータ本体。これは当たり前ですね。X68000シリーズなら，どの機種でもOKです。もちろんPC-9801でもできますが，Oh!Xの読者で，いまさらPCシリーズを買おうという人はいないでしょう。

2） MIDIボード

このインタフェイスボードがコンピュータミュージックの要です。先ほど述べた「コンピュータによって設定されたテンポにおける，何小節目の何拍目にどのキーがどのくらいの強さで押されたかというデータ」というのは世界共通規格のフォーマットに従って，コンピュータと楽器との間でやりとりされます。この世界共通規格のことをMIDI規格というのです。そして，MIDIボードはX68000と外部のMIDI楽器との通信を可能にするものなのです。

このボードにはシャープの純正品（CZ-6BM1）とシステムサコム製（SX-68M）の2種類が市販されています。このどちらでもOKですといえば，安いほうのSX-68Mを買うことでしょう。SX-68MでできなくてCZ-6BM1でできることは唯一FSK同期といって，これはMIDIとアナログ録音のテープレコーダとを同期させるものですが，よほどのマニアでない限り99％使いません。

3）外部MIDI楽器

コンピュータミュージックの大きな落とし穴がここにあります。すなわち，記録されるデータが音ではなく楽譜であるということから，曲を再生するためには演奏のための楽器が必要ということなのです。そして，もしプロの音楽家がスタジオの楽器をたくさん使って凝りに凝った作品を演奏したとしたら，それを再生する私たちもまったく同じ楽器をすべて揃えないと同じ演奏を再現することはできないのです。それでは世界共通規格というのはおかしいではないかと文句をいう人もいるでしょう。いえ，ちょっと待ってください。MIDIではデータとしてやりとりされるのは楽譜としてのデータなので，楽器が違っても同じ楽譜にしたがって正確に演奏されます。この点では世界共通規格なのです。ところが，楽器の音色は共通ではないので，いちおう音は出て，曲にもなっていたとしても，楽器が違うと曲の聞いた雰囲気は再現できないということなのです。

前置きが長くなりましたが，コンピュータミュージックで外部から持ってきたデータを再生するときにはどの楽器で再生することが前提になっているか気をつけなければならないのです。その点，X68000を含めいまのパソコン向けのデータはほとんどすべてローランド社のMT-32，CM-64，CM-32シリーズの対応になっています。コンピュータミュージックをこれから始めるのであれば，ローランド社のCMシリーズが絶対的にお勧めです。ほかの楽器を買ってしまうと，CMシリーズ用の曲を再生するときに自分で音色を設定しなければならず，しかも曲のニュアンスが完全に再現される保証はありません。

もちろん自分で演奏データを作成し，自分だけのシステムで音楽しようというミュージシャン指向の人は，最近の電子楽器はすべてMIDI規格ですから，自分の好きな機種を買ってかまいません。楽器店に行って実際の楽器をとにかく触りまくるのがいちばん。ローランドのMT，CMシリーズはすでに持っていて，さらに楽器を増やしたいという人でも，自分の買いたいものであればなんでもOKです。

ただし，楽器の選び方として，音源モジュールというそれ自身では演奏機能がないものと，キーボードなど演奏機能のあるものとがあることを注意しておきましょう。演奏データを入れる方法には，X68000のキーボード（あるいはツールによっては画面上のアイコンやソフトキーボード）のほかに外部楽器による自分自身の演奏をリアルタイムで録音する方法があります。

リアルタイム録音はある程度楽器が演奏できる人に限られますが，中学や高校の授業でオルガンかリコーダーを習ったと思います。そのときのことを思い出して楽器演奏に再挑戦してみるのもよいでしょう。そういった意味ではCMシリーズコンパチのキーボード（というよりは，以前YAMAHAのMSXパソコンにあったような，CMシリーズにオプションで付けられる音源なしのMIDIキーボード）を格安で提供してほしいものです。

4）音楽ソフトウェア

以上でハードウェアはすべて揃いました。あとはこれらのハードウェアをコントロールするためのソフトウェアが必要です。X68000上で走るコンピュータミュージック用のソフトウェアもずいぶん揃ってきて，目的によって選べるようになってきました。そこで，次に目的別にコンピュータミュージックを楽しむ方法を解説していこうと思います。

MIDIサウンドの楽しみ方

1）聴く

「音楽する」ことは，もちろん聴くことから始まります。コンピュータミュージックでもふつうのCDと同じようにデータ曲集を持ってきて，その曲を聴く楽しみ方があります。とりあえずお勧めするのは，Musicstudio PRO-68K（ver.2.0が発売になりました）を1本買ってきて，その中のサンプル曲を聴くことです。あるいはMu-1でもOKですが，将来ステップアップすることを考慮して，リアルタイム録音のできるMusicstudio PRO-68Kのほうがよいでしょう。

また，Musicstudio PRO-68Kには別売のデータ曲集が揃っているので，1枚ずつ買い足していくこともできます。どれも第一級のプロミュージシャンによるものですから，曲の質は保証付きです。私はなんといってもMusicstudio PRO-68Kのデモ曲としてついてきた国本佳宏氏による"Ikejiri slapstick"という曲が一番のお気に入りで，最初にこの曲を聴いたときはあまりの感動に1日中かけっぱなしにしていました。特にCM-64対応版はLA音源に加えてPCM音源のパートが曲に厚みを加え，さらにCMシリーズからドラムパートに加わった効果音がたいへんうまく使われていて，最高の傑作です。

このように，ミュージックツールから揃えていくのが基本でしょうが，私としてはむしろMIDI対応のゲームソフトを買ってくることをまず最初にお勧めします。ゲームソフトのBGMでも，最近のソフトウェアハウスでは専門の作曲スタッフが内容の高いものを作るようになってきています。ゲームソフトではただ聴くだけでなくまずはゲームそのものが楽しめるようになっているので一石二鳥ともいえましょう。

また，ゲームソフトによってはミュージックドライバと曲データとをゲームプログラムと切り離せるようになっているので，ゲームソフトを立ち上げなくてもBGMが楽しめます。実は今，私はスーパーハングオンのBGMをかけながら，付属のワープロでこの原稿を書いているところなのです。

あとはOPMDとOPMファイルの組み合わせでも気軽にコンピュータミュージックが楽しめます。OPMDはOh!Xの付録ディスク（1990年6月号）に入っていましたし，OPMファイルは電脳倶楽部を定期購読していれば毎月必ず2～3曲のペースでライブラリが増えます。ただ，MIDI音源対応のデータはほとんど掲載されていないので，同じデータを使ってMIDI楽器で演奏させるときは音色の選択は自分でやらなければなりません。

2）演奏データを打ち込む

「聴く」の次のステップとしては「演奏する」になるのですが，演奏するにも大きく分けて2つのパターンがあります。

まずは，演奏情報を音符ごとに音階，音長といった楽譜データとして1ステップず

つコンピュータに入力し，そのデータに従ってコンピュータに演奏させる方法で，これをステップ録音といいます。それに対して，実際に楽器をリアルタイムに演奏したものを同時にデジタルデータに変換してコンピュータに記録し，あとからそのデータに従ってコンピュータに演奏させる方法をリアルタイム録音といいます。ここで「演奏データを打ち込む」といった場合は，ステップ録音のことをいっているのだと思ってください。

さて，ステップ録音のために最も安あがりな方法はOPMD.XあるいはMUSICDRV.Xといったデバイスドライバを用意することです。これらはX-BASIC上のMMLやOPMファイルによるOPMDRV.X用の演奏データをまったくそのままの形で使えるようになっています。Oh!X LIVEに載っているデータをそのまま打ち込んでみるのもよいでしょう。ただ，この場合もMIDI音源対応のデータはほとんど掲載されていないので，注意が必要です(今月はMIDI特集ということでMIDI対応のデータも掲載しています)。

雑誌に掲載されているMMLデータではなく，自分の好みの曲を楽譜から演奏データに落としたいときにはいちいち自分でMMLに変換して打ち込んでいくのはお勧めできません。そもそも，楽譜なんて読めないよといって最初から自分で楽譜を見てデータを打ち込むのをあきらめてしまう人もいるかもしれませんが，そういう人でもとっておきのツールがあります。

それはMUSIC PRO-68K［MIDI］です。このツールは楽譜ワープロともいえるもので，起動すると画面には白紙の五線譜が現れます。あとはマウスやキーボードを使って，音符や演奏記号のアイコンを譜面上にはりつけていくだけで，演奏情報が書き込めてしまうのです。これならば，楽譜の中の個々の記号の意味を十分理解していなくても楽譜を丸写しさえすればすべてＯＫなのです。楽譜からただひたすら機械的にデータを打ち込んでいたような人にはぜひ試してみてほしいソフトウェアです。

3）演奏を録音する

これはすなわちリアルタイム録音のことです。これまでは演奏を録音するといえば，テープレコーダに録音して，あとから聴き直すという形でしたが，コンピュータミュージックでは最初にも述べたとおり，楽譜の形で記録します。

リアルタイム録音をするには，Musicstudio PRO-68Kを使うしかありません。Musicstudio PRO-68Kの操作性は基本的にテープレコーダと同じです。画面上にテープレコーダのそのものが映し出されるので，そのままRECスイッチをクリックして録音待機状態にし，次にPLAYスイッチをクリックするとレコーダが回り始めます。

このとき，メトロノームをセットしておくと内蔵音源あるいは外部MIDI音源からテンポをカウントするメトロノーム音が出力されるので，それに合わせて演奏してください。録音が終わったらSTOPをクリックして録音状態から抜け，カウンタをリセット（巻き戻しに対応）したのちに改めてPLAYをクリックすると録音された演奏を再生することができます。

これだけでは普通のテープレコーダに録音するのと変わりないように思われますが，Musicstudio PRO-68Kでのリアルタイム録音が特に優れている点を列挙してみましょう。

●録音と再生とでテンポが変えられる

初心者にとってコンピュータでのリアルタイム録音で最も便利な機能はこれでしょう。リアルタイム録音ではメトロノームをセットしてそれに合わせて演奏すると書きましたが，たとえ４分音符が１分間に150以上の速い曲でも録音のときには80ぐらいまで落として設定してやれば，メトロノームもゆっくり刻んでいくので余裕をもって演奏することができます。そしていざ再生するときにテンポを150にまで上げてやれば，どんな初心者でも速弾きのテクニシャンになれるのです。もちろんテンポを自由に変えても音はおかしくなりません。というのも，あくまでも１音１音が音符データとして記録されているからです。

●録音する楽器を選ばない

もしあなたがピアノしか弾けないのに，エレキギターのかっこいいソロを決めたいとしましょう。今からギターを練習しますか？　コンピュータミュージックならばピアノの手弾きで録音しておいて，あとから音色を変えてやるだけでＯＫです。音源モジュールは１台でピアノからギター，トランペット，ドラムにいたるまであらゆる音を出すことができます。

私は最近ウインドシンセというMIDI対応の管楽器に凝ってます。演奏の方法は普通の縦笛といってよいのですが，MIDI出力で外部キーボードからも音を出したりしています。極端な話，縦笛の各音階にスネアドラムやバスドラム，シンバルなどの打楽器音を割り当てて，笛でドラミングなんてこともできてしまうのです。

●録音したあとに部分的に修正できる

人間誰しもちょっとしたミスはつきものでしょう。リアルタイムに演奏して１発で決まらなかった場合，普通のテープレコーダでは最初からやり直しです。ところがコンピュータで録音すると録音データを部分的に修正することが可能です。ひとつ音を間違えただけでも，そこだけ直せばあとは知らん顔なのです。それだけでなく，１曲のうち簡単なところだけ手弾き，難しいところはステップ録音などという細かい技もできてしまいます。

●多重録音ができる

普通のステレオテープレコーダでは片面２トラックずつしか録音できません。Musicstudio PRO-68Kならこれだけで24トラック，要するに24回繰り返せば独りで24重奏までこなせるのです。もちろん全パートステップ録音でもOKです。高級なマルチトラックレコーダは24トラックのものだと数百万円してしまい，とても個人で使えるようなものではありません。

ステップアップに向けて

これからコンピュータミュージックを始めてみようという人ならば，これまで述べたような手順で，とにかくサンプル曲を聴いてみることと自分で雑誌や楽譜の演奏データを入力してみることまでを目標にしてみましょう。これだけでも十分楽しめると思います。

そして，よほど古い機種でない限り最近のMIDI楽器はX68000本体内蔵のＦＭ音源にまさる厚みのある音を出してくれますから，まずは音のよさを楽しんでください。最初はゲームソフトのBGMやOPMファイルから始めるのが一番よいでしょうが，できれば早いうちからミュージックツールを揃えておくとよいでしょう。

MIDIシステムのステップアップには機材を揃えること（ハードウェア）とツールを使いこなすこと（ソフトウェア）の両面が考えられますが，これから始めようという人にはハードウェアは最低限にとどめておいて，ミュージックツールを駆使して音楽する方向に向いていってほしいと思います。ツールを使ってどうコンピュータミュージックを表現していくか，これにはやはりある程度の慣れが必要ですが，それだけに作品の出来上がりは各人の感性に委ねられているので，コンピュータミュージックの真髄が楽しめるからです。

楽譜入力からリアルタイムアレンジまで

ミュージックツール実践活用

Shimada Atsushi 島田 淳史

コンピュータに詳しい人なら自らMIDI環境を作ることも可能でしょうが，まずMIDIの力を知るためにゲームソフトのBGMを聴いたりMIDI用ソフトでデータを入力してみましょう。ここでは市販ツールを使って何ができるかを紹介します。

コンピュータミュージックといっても，X68000本体しか持っていない皆さんはOPMD+MML（OPMファイル）というパターンがほとんどだったのではないでしょうか。しかし一方，MUSIC PRO-68K［MIDI］やMusicstudio PRO-68K ver.2.0といったいわゆるミュージックシーケンサソフトも充実してきて，DTM（Disk Top Music）ではほぼ完成の域に達しているMacintoshの世界を彷彿とさせるようになってきました。

コンピュータミュージックを志す以上，私たちもこのようなミュージックツールを使いこなしてみたいと思うはずです。そこで，この記事ではミュージックツール実践活用と題して，MIDI機器のセットアップから実際に，MUSIC PRO-68K［MIDI］やMusicstudio PRO-68K ver2.0を使っての曲作りとアレンジにいたるまでをドキュメント風に解説してみることにしました。

MIDI機器のセットアップ

システムの構成図は図1のとおりの基本形です。パソコンショップではお手軽MIDIセットなどという名前がついて販売されているものです。セットを買ってきたら，まずMIDIボードを本体に装着し，CM-64付属のMIDIケーブルをボードのMIDI OUTからCM-64のMIDI INにつなぎます。ここを間違えるとまったく音が鳴りません。また，MIDIというのはあくまでも演奏情報の通信に使われているので，実際に音を出すには，CMのステレオ出力からオーディオシステムに接続してやるのも忘れてはなりません。

接続完了のチェックには，MUSIC PRO-68KあるいはMusicstudio PRO-68Kのサンプル曲を演奏させてみるのがよいでしょう。まず，MUSIC PRO-68Kシステムディスクを立ち上げます。画面上のディスクアイコンをクリックするとファイルリストが表示されますので，DEMO1.SCOをロードしてみます。ファイルメニューから出たのち，今度はスピーカーアイコンのプルダウンメニューで全曲演奏を選ぶと演奏が始まります。ここで音が何も出なかったら要チェックです。まず，演奏状態にしてからCM-64正面パネルのMIDI MESSAGEランプを見てください。ここが点灯しなければ，MIDI信号自体が送信されていません。MIDIボード，ケーブルなどの接続をもう一度確認してください。あるいはMIDI用のデバイスドライバが登録されていないこともあるかもしれません。ランプが点灯して，なお音が出ない場合はオーディオシステムの接続ミスです。そちらのチェックは各自にお任せします。

MUSIC PROで楽譜入力をしてみよう

MUSIC PROは楽譜ワープロといわれているツールで，画面上の五線譜に音符を書き込んでいくだけで演奏データを入力できるものです。MMLで記述したデータはそれだけ見てもどこからどこまでが1小節分かなどということがわかりづらく，入力途中で頭が混乱しやすいのが難点でした。楽譜を用意して演奏データに落とすとき，そのままの形で写せるのはとてもわかりやすくて効率も上がります。

では，さっそく使ってみましょう。仮にここに，YMOのRYDEENの楽譜があったとして（ちょっと古いですが），これを入力してみましょう。全部で7パートあるのですが，メロディパート，アドリブパート，リフレインパート，ベースパート，リズムパートの5パートぐらいにしぼってみます。また，全体の曲構成ですが，メイン―アドリブ1―メイン―アドリブ2―エンディングの5ブロック構成にします。

まずはパート設定から始めます。文書アイコンを選択すると設定メニューが現れますので，すべてのパートについてその音域（ト音記号かヘ音記号）や発音音数などを設定します。ここでは特に各パートの発音数に気をつけましょう。というのも，全体で16音というのが決まっているからです。まあ，RYDEENの楽譜の場合，メロディパートとリズムパートが同時に3音ずつ必要なだけで，あとのパートは単音なので，音数にはかなり余裕があります。

パート設定が終了すると五線譜が設定したとおりに書き換わるのが確認できたでしょうか。そうしたら今度は全体のテンポ，調性，拍子を設定しますが，それぞれメトロノームアイコン，調性記号アイコン，拍子記号アイコンをクリックするとメニューが表示されますので，必要なものを選んで設定していきます。

いよいよ音符を入力していくのですが，各パートごとにまずMIDIチャンネルの設定と音色（プログラムチェンジ）の設定す

図1 MIDI基本システム例

MUSIC PRO-68K[MIDI]

Musicstudio PRO-68K

るのを忘れないようにしてください。最初はリズムパートから入れていくのが通例です。リズムパートで気をつけなければならないのが，各ドラム音源はひとつずつある音階に対応させられているという点です。したがって，楽譜のリズム譜を見て，スネアドラムやバスドラムといったそれぞれのドラム音にはその対応する音階の音符をMUSIC PROの五線譜に書き込まなくてはなりません。

音符もアイコンで必要な音長の音符を選択し，譜面に置いていくだけでOKです。もし間違えたとしても，ケシゴムアイコンで1音符単位やブロック単位で修正が可能なので，きれいな楽譜を書くことが可能です。まさに楽譜「ワープロ」の面目躍如といったところでしょう。

さて，リズム譜をよく見ると4小節単位ぐらいで同じパターンが並んでいることがよくあります。このときは，ブロックコピーを活用しましょう。ハサミアイコンでコピー元のブロックを切り出し，コピー先を指定してやるだけでコピーできてしまいます。それだけでなく，繰り返し記号もしっかりサポートされているので，たいへん見やすい楽譜を書くことができますし，たいていの楽譜なら全部ベタ写しも可能なほどの実力をもっています。

次にはベースパートを入力していきましょう。ベースパートも同じパターンが4小節単位ぐらいで続いていくので，コピーを活用すれば，あっという間に完成です。リフレインパートは16分音符の連打なので，楽譜に書き込むときに配置に注意しなければなりません。見栄えはなんかごちゃごちゃしてしまいましたが，演奏させてみると正確に演奏してくれるのでひと安心です。アドリブパートはライブなどでギタリストやキーボーディストが文字どおりアドリブ演奏を決めるところですが，楽譜にはそのアドリブ演奏が譜面として載っているので，一応そのとおりに入力しておきましょう。

このようにしてリズムパート，ベースパート，リフレインパート，アドリブパートと入力してきました。最後にメインのメロディパートも同じように入力します。

ところで実際のところ，私はメロディパートはMUSIC PROでは入力しませんでした。というのも，私自身は最初に示した基本システムのほかにMIDI対応のシンセサイザキーボードを持っているので，メインメロディはリアルタイム録音したのです。要するに，MUSIC PROで入力した残りのパートをバックにカラオケとして流しながら，それにあわせて自分で演奏したわけです。ですから，メロディパートはまったく空白のままSCOファイルを完成させてしまったことになります。

皆さんは基本システムしか持っていないと思いますから，メロディもMUSIC PROで入力してください。全部入力し終わったら，全体を演奏させて，入力ミスなどないように丹念にチェックしておきます。まだ，音色や音の強弱などのバランスは取れていないと思いますが，この時点では楽譜としての音階と音の長さとを正確に入力することに重点を置くだけでかまいません。MUSIC PROには強弱記号もサポートされているので，これを使ってみてもよいと思いますが，あとでリアルタイムアレンジをするときにそのデータが邪魔になることもあるかもしれません。

さあ，これで自分が納得すれば，その時点でSCOファイルの完成です。次のリアルタイム録音の説明はさっと読み流して，その次のリアルタイムアレンジのほうに進んでください。

Musicstudioでリアルタイム録音

このようにしてMUSIC PROで入力した各パートのSCOファイルはMUSファイルに変換してセーブしておきます。このMUSファイルを今度はMusicstudioのSNGファイルに変換してやるのです。変換のしかたは極めて簡単です。Musicstudioを起動してから目的のMUSファイルのアイコンをダブルクリックしてやると，「MMLファイルをSONGデータに変換します。よろしいですか？」と聞いてきますので，OKを選んでください。新しいファイル名を入力するとあとは自動的に変換してくれます。

この新しいSNGファイルを改めてロードしてください。やはり，目的のアイコンをダブルクリックするだけです。ロケーションコントローラのPLAYをクリックしてMUSIC PROで入力したRYDEENの各パートが流れてくるのを確認してください。このとき，各トラックのSOLOスイッチを入れるとそのトラックだけ演奏されますので，パートごとのチェックも簡単です。

さていよいよリアルタイムアレンジですが，その前に，実際に私が行ったリアルタイム録音も参考までに説明しておきましょう。まずMIDIキーボードの出力チャンネルをCH2にセットします。私の場合はトラック2があいているので，その受信チャンネルがCH2であることを確認してからRECスイッチをクリックします。その状態でロケーションコントローラのPLAYスイッチをクリックすると，有名なハイハットのイントロからRYDEENが始まります。そのバックに合わせて，演奏するのです。

メロディパートは2つのアドリブパートの間奏をはさんで3回同じフレーズを弾くことになります。私は勢いで，3回とも手弾きしたのですが，1回だけにしておいて止めてしまい，あとはステップエディットで同じフレーズをコピーしてしまうという手もあります。

残念ながら，3回のフレーズの中で，1回ミスをしてしまいましたが，Musicstudio PRO-68Kではそのミスをした小節だけ録音し直すことができます。これはパンチイン・パンチアウトというテクニックなのですが，録音を入れ替えたい部分の始めと終わりの小節番号をロケーションコントローラにセットしておいてから，先ほどと同じようにリアルタイム録音の状態にすると，たとえ最初から弾いていったとしても，今度は指定した部分だけデータが入れ替わってくれるのです。

さて，リアルタイム録音が終わって，全パートを聴き直してみると，どうも手弾きの部分だけときどきタイミングがずれているようです。そういう場合には，クォンタイズが有効です。

クォンタイズというのは，リアルタイム録音のデータのタイミングのバラツキを補正してくれる機能です。Quantizeメニュー

を選ぶと，クォンタイズをかけるトラック，小節を聞いてきますので，今手弾きしたトラック2の全小節を指定し，最小単位を32分音符ぐらいに設定してOKをクリックします。すると，32分音符のタイミングですべてのずれが補正されます。改めて聞き直してみると，ぎこちないタイミングがよくなっているのがわかるでしょう。以上で全パートのSNGファイルが完成しました。

誰でもできるリアルタイムアレンジ

これまで述べてきたように，基本となるSNGファイルはMUSIC PROですべて作ってきました。これだけでも一応曲になるのですが，どうも各パートの音色や音の強弱のバランスがいかにも機械的にしか聞こえないものしかできません。やはり，ステップ入力で音楽を表現するのはかなり大変のようです。そこで，MusicstudioのMTR上で，全体を加工してみましょう。

まずは音の厚みを付けるためにトラック2～5のパートをトラック11～14，17～20に2重にコピーします。それにはTrack copyメニューを選び，2→11，3→12というように順番にコピーしていきます。すべてコピーし終わると同じパートが3つずつできることになります。

そこで，トラック11～14のMIDIチャンネルをCH11～14に設定します。そうすると，これらのパートはCM-64のPCM音源部から音を出すことになります。さらにトラック17～20はトラック番号の下のセレクタをすべてMIDIからFMに変え，チャンネルをCH1～8に設定します。

Musicstudioの新しいMUSICDRV.X ver.2.0はMIDIと同時に内蔵FM音源もドライブできるので，これによって，同じパートがFM音源からも出力されることになります(ただし，FM音源は1パート1音なので，パートの中の和音は正しく出力されない)。以上で，リズムパートを除くすべてのパートがLA音源，PCM音源，そしてFM音源の3つのユニットからユニゾンで演奏できるようになったのです。

これで相当音が厚くなったので，次に音色のバランスを取ってみましょう。それには，最初から演奏させながら，各トラックのプログラムチェンジを変えていくことです。CM-64の音色表から合いそうな音色を選んでは試しにセットしていきます。LA，PCM，FMと3種類音源があるのですが，すべて同じ系統の音に揃えてしまうとかえって，音が濁ったりして聴き心地が悪くなります。あるいは3パートもあるのにすべて同じ音だとまったく厚みが感じられないこともあります。

たとえば，ベースパートを例に取ると，3ユニットすべてシンセベースに設定しても，低音の響きが軽く，RYDEENならではのベースラインが全然はっきりしてきません。そこで，PCM音源は一応スラップベースなどにしておいて，LAのほうを思い切ってパイプオルガンなどにしてみると意外とふくらみのある低音が響きます。

このようにいろいろ試行錯誤しては全体の音のバランスを決めていくのです。もしこれがベストというのが決まれば，ステップエディットで各パートの始めにプログラムチェンジ命令をセットしておくのを忘れないように。

次に各パートの音のバランスを決めていきましょう。それには，各トラックのボリュームフェーダを適当な位置にセットしていきます。たとえば，リフレインパートは音域が高いところにあって音が響きやすいので，ほかのパートよりも少し音を小さめにセットしてみようとか，バスラインの響きを重視して，バスパートの音量を少し高めにセットしようとか，各自の好みにしたがってアレンジしてください。また，パンポットなどもいじってみるとステレオ感が出てきたりするので，面白いと思います。

いずれにしても，この音の強弱は先ほどの音色を決めるのと互いに密接な関係にあって，音色によって音の強さ，あるいは音の強さによって音色が決まったりします。根気よくアレンジを繰り返して，全体の曲の雰囲気を仕上げてください。これも，各パートの始めのほうにメインボリュームの設定命令があると再生し直したときにすべて元どおりに変えられてしまうので，注意が必要です。

以上で，ほぼ完成です。最初にMUSIC PROで譜面入力したときに比べてだいぶ質感が上がったことでしょう。これと同じ作業をMUSIC PROあるいはOPMファイル上で行うのは結構手間がかかります。このように適切なツールを使い分けることによって，自分自身の音楽性も効率よく高められていくのです。

さて最後に，Musicstudioならではのリアルタイムエディットを紹介しましょう。このRYDEENのエンディングをフェードアウト（次第に音が小さくなっていって終わる終わり方）で決めたいと思います。

まず，Rec modeメニューをInternal＋Overdubにセットします。そして，データの入っている全トラックのグループスイッチをONにし，RECスイッチもONにします。ロケーションコントローラでフェードアウトを始める小節の頭にセットします。

全トラックのうち一番ボリュームの上がっているフェーダにマウスのポインタを合わせ，スペースキーを押すと内部ミキシング録音がスタートするので，マウスを合わせたフェーダをゆっくり下げていきます。すると全フェーダが一緒に下がっていくのがわかるでしょう。全体をボリューム0まで下げたら再びスペースキーを押して録音を止めて完了です。全パートにいまのフェーダの動きが記録されました。

このようにInternal＋Overdubの録音モードにセットして録音状態にすると，フェーダ，パンポット，プログラムチェンジなどがリアルタイムに動かしたとおりに記録されるので，いまのフェードイン・フェードアウトやクレッシェンド・デクレッシェンドなども感覚で操作することができます。これさえマスターすれば，気分はもうプロのミキシングエンジニアそのものです。

あなたも立派なミュージシャンだ！

コンピュータミュージックを「機械いじり」の延長から「音楽の演奏」へとステップアップしていく様子がわかってもらえたかと思います。もちろんMMLやOPMフォーマットでデータを記述していくのも立派な音楽です。むしろ，自分の表現したい曲の雰囲気をOPMファイルに落とすことのほうがはるかに難しいかもしれません。というのも，OPMファイルをそのまま眺めているだけでは，曲のイメージがなかなかつかみにくいからでしょう（西川善司氏ほどのストロングタイプになるとYコマンドの羅列を見ただけでも曲が口ずさめてしまうのかもしれませんが)。

しかし，どんな人でも足踏み，手拍子，鼻歌など，体全体で音楽を表現する能力は持ち合わせているはずです。その個人個人の感性をどうコンピュータミュージックのデータとして表現するか，それはいかにしっかりしたコンセプトと操作性を備えたツールと出会えるか，そしていかにそれを使いこなしていくかにかかっているのです。そういった意味で，ぜひ皆さんにもX68000に用意されたMUSIC PROとMusicstudioを使って実際に曲作りを試してみてほしいのです。きっとこれまでのMMLで音楽していたときとは違った感性が湧き出てくるに違いありません。

グラフィックチェーンプレーが可能

Musicstudio PRO-68K ver.2.0

Misawa Kazuhiko 三沢 和彦

X68000用のMIDI対応ソフトの第1弾であったMusicstudio PRO-68Kがバージョンアョプして帰ってきました。内蔵音源にも対応，楽しいグラフィックチェーンプレイなど機能を大幅にアップしての登場です。

このMusicstudio PRO-68Kはマルチトラックレコーダ（MTR）感覚の操作性で，マルチパートからなるオーケストラの多重録音・演奏を実現した本格的ミュージックシーケンサソフトです。しばらく前にマイナーチェンジをしてver.1.1が発売されましたが，今回のバージョンアップは機能，操作性ともにかなり完成度の高いものとなっています。

まずは起動させてみよう

MIDIボードを拡張スロットに差してからMusicstudio PRO-68K ver.2.0を立ち上げてみましょう(MIDIボードがなければ使用できません)。MTRのウィンドウを開けてみると，まずは旧バージョンに比べて大変カラフルになって画面のセンスが格段に上がったのに気づくと思います。この画面はなんとサン・ミュージカル・サービスから発売されているMu-1とそっくり同じです。Musicstudio PRO-68KとMu-1はどちらもサン・ミュージカル・サービスによって開発されたものなのです。ですから，このMusicstudio PRO-68K ver.2.0もMu-1で特徴的な機能はすべて受け継いでいます。

さっそく音を出してみましょう。付属のデモ曲はRolandのCM-64，CM-32(MT-32)に対応しているので，皆さんもまずはこれらの機種を揃えることをお勧めします。MIDIケーブルでしっかり接続してからデータディスク上のデモ曲をロードしましょう。操作性抜群のミュージックシェルではマウスでFileメニューからLoad songをセレクトすると，ファイルウィンドウがオープンするので，デモ曲のアイコンをダブルクリックするだけでOKです。

あとはロード完了とともに自動的にオープンするロケーションコントローラのPLAYスイッチをクリックすると曲がスタートします。このロケーションコントローラもカセットデッキの操作パネルにそっくりです。また，Mu-1のウリだった98曲ランダムチェーンプレイももちろんサポートされているので，普通のCDプレイヤーとまったく同じ感覚で操作できます。

内蔵FM音源・AD PCMもドライブ

Mu-1ではPC-9801用ミュージくんのソングデータをファイルコンバートする機能が付いていました。この機能によってランダムチェーンプレイなどに使えるデータライブラリが飛躍的に増えました。しかし，ミュージくん用に市販されている曲データは内容が貧弱というかレベルが低く，私自身もあまり得をした気分にはなれませんでした。それよりは，電脳倶楽部などでかなりの量が普及してきたOPMファイルのほうが数段利用価値が高いと思っていたのです。その期待に応えるかのように，このMusicstudio PRO-68K ver.2.0にはMIDIボードだけでなく内蔵FM音源・AD PCMもドライブできるMUSICDRV. X ver.2.0が付いてきました。

しかもこのMUSICDRV. X ver.2.0はあのOPMDコンパチブルなのです。ですから，いままでOPMDで演奏させてきたデータがそっくりそのまま生かせるのです。PCMデータもこれまでの12音から128音一括ロードが可能になり，OPMD用のAD PCMデータコンフィグレーションファイルもそのままのかたちで使えます。

そして，OPMファイルをMusicstudio PRO-68Kにロードするには，ファイルウィンドウをオープンして，OPMファイルのアイコンをクリックするだけで，自動的にファイルコンバートを実行してくれます。あとはそのままセーブしてやれば，MIDI専用のSNGファイルとまったく同じようにデータエディット，ミキシングなど思いのままにできるわけです。

さて，OPMファイルからコンバートしたSNGファイルをロードすると，MTRの各トラックのレベルメーターの下の表示がMIDIからFMまたはPCMに変わっているのに気づくと思います。このver.2.0からは各トラックの演奏データを自由にMIDI，FMあるいはAD PCMのどれかに割り振ることができるようになったのです。ですから，FM音源用のトラックデータをあとから別のMIDI楽器に出力を切り替えてやることもできるようになりました。もちろんこれはMIDI楽器をあとから買い足したときのことですが，もともとX68000内蔵のFM音源でもひと昔前のDX-21，DX-100といったYAMAHAのシンセサイザと同等の性能なので，音源としては侮れない実力を持っています。ですから，最初からMIDI楽器がAD PCMのドラムパートとともに1台揃っていることに変わりないと考えてよいでしょう。ということは，もし予算の都合でMIDI楽器が手に入らないという人でも，Musicstudio PRO-68K ver.2.0は十分楽しめるソフトウェアなのです。

MTV顔負けのグラフィックチェーン

ランダムチェーンプレイは演奏データだけではありません。なんとグラフィックデータもロードして画面表示させることができるのです。しかも，そのグラフィックデータは皆さんお馴染みのPIC形式まで対応しています。

実際の手順を説明しましょう。まず，単純にグラフィックデータを表示させるには，データのあるディスクドライブのウィンドウをオープンして目的のファイルアイコンをクリックしてやるだけでそのまま画面が切り替わり，グラフィックが表示されます。チェーンプレイを行うには，ウィンドウメニューのChain playをセレクトすると，チェーンプレイのウィンドウがオープンしますので，その状態でプレイさせたいファイルのアイコンを順次クリックしていきます。そうするとウィンドウ上にファイル名が登録されていくので，適当なところでチェーンプレイをONするとスタートします。このチェーンプレイのウィンドウ上には，

SNGファイルとグラフィックファイルを混ぜて登録することができ，曲ごとに画面を切り替えていくようなMTV（ミュージックテレビ）そのものが実現できるのです。

PICファイルもOPMファイルと共にかなり質のよい作品が出回っていますので，その日からでもMTVもどきのプレイが楽しめるのです。

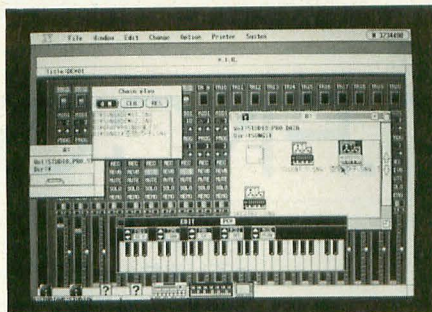
ステップ入力もスピーディに

デモ曲や他の人の打ち込んだ曲を聴くのに飽きたら，自分でデータを打ち込んでみようと思うことでしょう。Musicstudio PRO-68KではやはりMTR本来の使い方としてのリアルタイム多重録音がお勧めですが，BASICのMMLやOPMファイルで直接データを記述していた人にはまずステップ入力が手頃です。

ステップ入力は音の高さと長さを1組の数字データで表して，そのデータを列につなげて演奏情報とします。特に音の長さの表現として4分音符の長さを適当なステップ数で区切り，その1ステップを単位として各々の音符の長さを表すところからステップ入力といいます。ですから，BASICのMMLやOPMファイルで演奏情報を記述するのと基本的には変わりありません。しかし，このステップ入力がMMLやOPMファイルよりも優れている点は小節番号，拍子番号，ステップ数が明確に管理できるという点です。

MMLでは，文字列が順にただ並んでいるだけなので，もし音符を打ち間違えると全体の楽譜の中でどこが狂っているのかあとから探すのにひと苦労です。特に音の長さについては，最初から順に足し合わせていくだけなので，どこか1カ所長さを間違えると以後は全部ずれてしまいます。

一方，Musicstudio PRO-68Kのようなツールのステップ入力では，ひとつの音を入力するごとにその音の絶対的な位置，すなわち，何小節目の何拍目の音かという情報がわかるようになっています。ですから，音の長さを間違えて入力しても，たいていの場合次の小節でステップ数が合わないことがわかるので，間違いが発見しやすいのです。さらに，楽譜全体を入力したあとに途中の1音を修正するときも，目的の音の位置を探すのがとても楽になります。

さて，Musicstudio PRO-68Kでステップ入力する際に音の長さは基本的に画面上のアイコンで指定していきます。しかし，音の高さを指定するにはソフトキーボード，X68000のキーボード，そして外部MIDIキーボードの3通りの方法があります。

チェーンプレイが面白い

まず，ソフトキーボードは画面上に表示される鍵盤で，各々のキーをマウスでクリックしていくとその位置の音を出すことができます。X68000キーボードの場合はアルファベット入力用のキーに各音が対応しているので，楽器のキーボードと同じようにキーを押すことによって音を出します。MIDIキーボードの場合は普通に演奏するのと同じようにキーを押さえていくと音の高さのデータが入力できます。このとき，和音も一度に入力できるので非常に便利です。CMシリーズなどの鍵盤がついていない音源モジュールしか持っていない人は，X68000のキーボードから入力する方法が慣れると速いのでお勧めです。

このような音符データ入力は慣れるまでが結構大変で，使っているツールの出来の善し悪しが効率を大きく変えてしまいます。この点，Musicstudio PRO-68Kの操作性はかなりよくできていると思います。また，ver.2.0からは，マニュアルの記載がメニュー順でなく，操作別に改訂されたので，初めて使う人でも迷うことがなくなりました。アイコンやメニューの説明も必要な場所でそれぞれ説明されているので，マニュアルの説明に従って実行しながら操作方法を覚えていくことができます。

楽器が不得手でもリアルタイム演奏を

ステップ入力で打ち込んだデータでいつも苦労するのが，リアルタイムのドライブ感だと思います。リアルタイムの演奏を録音した場合では，自然と音の強弱による演奏の表情が出ます。それに比べるとステップ入力の場合では，演奏の表情を実際に聴いた感じでカッコよく再現するにはこまめにデータの数字を変えていくしかありません。そこで，威力を発揮するのがリアルタイムエディット機能です。

これは，MTR画面上で各トラックのボリュームやパンポット，プログラムチェンジなどを曲を演奏させながらリアルタイムに操作して，なおかつその操作内容を録音データとして残すことができる機能です。この機能はMu-1から追加され，Musicstudio PRO-68Kの旧バージョンではMIDIケーブルを外付けする必要がありました。新バージョンからはアレンジしたい曲をロードしたあとにレコーディングモードをInternal＋Over dubに設定して録音待機状態から曲をスタートさせるとリアルタイムエディットに入ります。特にフェードイン・フェードアウトを複数のトラックにわたって行いたいときは，MMLでは気の遠くなるような細かい設定が必要ですが，Musicstudio PRO-68Kではグループフェーダ機能をリアルタイムエディット時に使うとひとつのボリュームフェーダを操作するだけで簡単に実現できます。

MIDI楽器だけでなく内蔵音源も利用できる

その他，機能はもりだくさん

これからMIDIを始めてみようという人や，これまではMMLしか使わなかったという人にはこれまで説明してきた機能だけで十分使いごたえのあるものとなっているでしょう。特にMUSICDRV. X ver.2.0が付いてからは外部MIDI音源がなくても十分楽しめるソフトになっています。

残念ながら，このソフトの機能はこれだけの誌面ではまだまだ説明しきれません。実際，Musicstudio PRO-68Kの本当の実力はリアルタイム録音にあるのですが，今回のレポートではまったく触れられませんでした。リアルタイム録音とステップエディットの組み合わせはプロレベルの音楽製作にも十分応えられるほどの機能を持っています。このように，本当の初心者からマニアックなユーザーまで，幅広くカバーしているソフトですので，このMusicstudio PRO-68K ver.2.0をひとつ揃えておかない手はありません。ぜひお勧めのソフトです。

●Musicstudio PRO-68K ver.2.0
X68000用　5"2HD版2枚組　28,800円(税別)
シャープ　☎03(3260)1161

MUSICDRVの活用

MUSIC1.FNCで遊ぶ

Nakano Shuichi 中野 修一

「パソコンでMIDI」といってもまだアプリケーションは出揃っていません。ここでは「楽器を制御する」ところから始めてみます。BASICを使って，できるだけ簡単なサンプルで「MIDIを遊ぶ」ことにしましょう。

MIDIがあればさまざまなデジタル楽器を制御することが可能になります。これはどんなコンピュータでも同じことです。プロも使うようなコンピュータミュージックではMacintoshやATARI STなどの海外パソコンが高い評価を得ています。こういった本格的なコンピュータミュージックの世界ではリアルタイム録音が中心で，コンピュータはもっぱらシーケンサ，というよりMTR(マルチトラックレコーダ)として使われています。それに対して日本でのコンピュータミュージックはどちらかといえばステップ入力，楽譜入力などが重視されているようです。

大学祭でMUSICDRVを使ったMIDI楽器の演奏を行った西川善司氏のエピソード。十数曲にわたる曲を「ベンドからなにから全部ステップ入力だから……」と話して音楽野郎の目を丸くさせたといいます。まあ，MMLによる表記もステップ入力の一種です。ふつうにシーケンサを使っている人にとっては全部ステップ入力というのは想像もつかないことでしょう。

コンピュータとMIDIを使えば演奏技術がなくてもプロと同じ楽器を使いこなすこともできます。近頃はミュージ君やミュージ郎のデータ集なども出回っています(X68000ではMu-1で演奏可能)。MIDI対応のゲームも増えています。

さて？

少なくとも私は，プロミュージシャンを目指しているわけではありません。楽器は持っていても恥ずかしながら弾きこなしているとはいえませんし。なにもプロまがいの音楽活動をやりたいというわけでもなく，かといって一人前に楽器を揃えながらただ他人の作った曲を聞いているだけではもの足りないもの。

どうも「MIDI＝音楽」というものに対して身構えてしまうと自由な発想が途絶えてしまうような気がします。どんな便利なものも「遊べ」なければ，意味がありません(私見です。念のため)。

ここでは1月号の付録ディスクに入っているMUSICDRVを使います。そして「MIDIを活用する」ではなく，「実用になることをする」でもなく，ひたすら「MIDIで遊ぶ」ことを考えてみましょう。

どうやって扱うか？

それではMUSICDRVを組み込むところから始めましょう。展開されたディスクからMUSICDRVを探し，システムディスクまたはハードディスクのSYSディレクトリにコピーしたとしましょう。目的にあわせてCONFIG.SYSを書き換えます。

●MIDI中心に使う場合

DEVICE = A:¥SYS¥MUSICDRV.X /MOP

●OPMDの代わりに使う場合

DEVICE = A:¥SYS¥MUSICDRV.X /OPM /P450 /Y0

AUTOEXEC.BATで，

COPY BOS MIDI

を実行(ただしMIDIボードが必要)。BOSはOPMDで使用していたコンフィギュレーションファイルです。これでほぼOPMDと同じ演奏環境ができあがりました。

次はBASICにMUSIC1.FNCを組み込んでください。BASIC.Xが入っているディレクトリにMUSIC1.FNCをコピーし，エディタを使ってBASIC.CNFを書き換えます。

FUNC = MUSIC

という行を，

FUNC = MUSIC1

に打ち変えればいいでしょう。これでBASICからMUSICDRVの機能が使えるようになりました。

* * *

他機種のBASICと比べてX-BASICは使いにくいと思っている方，まずファイル処理を覚えましょう。Human68kではプリンタもFM音源もAD PCMもRS-232Cも正しく“ファイル”として扱われます。しかもランダムだのシーケンシャルだのといった面倒な分類はありません。

MUSICDRVを使うことによってMIDIもファイルとしてアクセスできるのです。さて，Oh!Xでは以前にもMIDIDRV.SYSというドライバが発表されており，それに

Rolandのチェックサム

MIの場合はインプリメンテーションチャートどおりにデータを並べてやればよかったのですが，MT-32やCM64では「アドレス」「チェックサム」で戸惑う人も多いはずです。手元にあるYAMAHAのQY10のインプリメンテーションチャートでもチェックサムが要求されることがありますのでわりと一般的な方式なのかもしれません。

最近のものはともかく，Rolandの初期のマニュアルではチェックサムの解説として不可解なことが書いてありましたので，Rolandのチェックサムはしばらく謎に包まれていました。最近のマニュアルでも「チェックサムはアドレス，データおよびチェックサム自身を加算した値の下位7ビットがゼロになる値」です。三沢氏が以前解析した結果，

X＝80H－(SUM) AND 7FH

という式を導き出しました。

アドレス(3バイト)を1バイトずつに分けて足し，データを1バイトずつ加えていきます。このときの合計がいわゆるチェックサムですが，さらにこの数字に加えたときに下位7ビットが0になるような値をチェックサムと呼んでいるようです。

Rolandのエクスクルーシブではシンセサイザのメモリ内とワークエリアを直接書き換えます(少なくとも概念としては)。どの機能がどのアドレスに配置されているかはマニュアルに記載されています。

MIDIのデータはすべて7ビット，すなわち7桁の2進数で扱うということを覚えておいてください。すなわちアドレスも1バイトあたり7FHを超えてはいけないのです。2進数で，

1111 1111 1111 1111B

というデータは，

11 0111 1111 0111 1111B

というふうにして扱います。サイズ(3バイトで表記)についても同じことがいえます。Roland関係のMIDI制御プログラムを作ろうという方は気をつけてください。

対応してMT-32シリーズとM1シリーズの音色エディタが発表されています。こちらもMIDIをファイルとして扱っているのでこれをMUSICDRVに対応させることは簡単にできそうですね。

実際，システムファイル名を“XMIDI”から“MIDIR”に変えるだけのことです。“MDC”を使った初期化の部分はデバイス名“MIDI”に(RE)を送ることで代用します。このようなRecieve Enableの処理を行わないと信号が読めないので十分注意してください。

要領さえわかれば，難物とされているエクスクルーシブメッセージも簡単に扱うことができます。演奏に直接関係ないこと，たとえば，楽器の音色をエディットしたり，メモリに持っているデータを表示させたりということはすべてエクスクルーシブメッセージで行われます。これは楽器ごとに仕様が異なります（逆にいえばこれ以外は全楽器共通）。

簡単なサンプルとしてKORG M１のエクスクルーシブメッセージを扱ってみましょう。まず，ModeChange( )という関数を作ってみました（リスト１）。配列ex( )にエクスクルーシブヘッダ（命令内容や適用機種を示すID情報），コマンド，データ，$7E_H$(EOX：エクスクルーシブの終わりを示す）を並べておきます。これが転送するメッセージ内容となります。あとはデータ部を書き換えて“MIDIB”（BはBinaryのB）に送信するだけです。内容はファイル処理の基本例といえます。

データを読む

まず，楽器から送られてくるデータを表示してみましょう。Musicstudioなどにある MIDIモニタを作ってみます。それにはMIDIからすべての情報が必要なので，まず，ファイルオープンで“MIDIB”を指定します。そこからデータを読み込んで表示するだけです。

MIDIではコマンドとデータの区別が非常に簡単ですからコマンドをみつけたときは改行動作を入れてわかりやすくしましょう。楽器によってはノートオフで$9n_H$を返さずにベロシティ（音量を０にすること）で音を消す機種もありますので，そのような場合はノートオフへの分岐条件を変更してください。

リスト２ではノートオンとノートオフで色を変えています。それでもどこの音がどこで終わったかはわかりにくいですね。同時に鳴っている音はさほど多くないはずですから，ちゃんとひとつの音が出てから消えるまでを追いかけてみましょう。

まず，最大で16音しかならないことにして(別に何音でもいい)，現在その音が出ているかをチェックします。配列ch( )にデータは格納されます。ノートオフがきたらそれと照らし合わせてやればいいわけです（リスト３）。

鍵盤を接続して適当に弾いてみましょう。画面の端にどの音階の音が出ているかが常に表示されるはずです。16音押さえるのは大変ですが，それぞれのキーオン/キーオフがちゃんと読み取れます。音階がわかればMIDI入力された信号の鍵盤表示のようなことは簡単ですね。興味のある方は各自で対応してください。

問題は17音目の処理が楽器の動作と違う場合があるということですが，新しい音を優先するか，いま鳴っている音を優先するかは楽器によって違います。このプログラムでは新しい音は無視されます。実用上は問題ないと思いますが，なんなら32音に変更するなりすればよいでしょう。

＊　＊　＊

さて，鍵盤表示は難しくなさそうだとわかりました。このようにデータが取り出せればあとは加工してMIDI OUTから流したり（プログラムの変換やフィルタリングができる），MML化してファイルに落としたりということが考えられます。

リスト４は昔発表したショートプログラムです。これはリアルタイムに与えられる音階データ（キーボードを鍵盤に見立てて弾く）をMML化して表示するものでした。これと同様にすればMML化は実現できるでしょう。元々ミュージックデータ入力を

リスト1

```
 10 /* MODE CHANGE
 20 char combination=0,  edit_prog=3
 30 char edit_combi=1 ,  global=4
 40 char program=2    ,  sequencer=6 :/* 5 ではない
 50 /*
 60 char ex(7)={&HF0,&H42,&H30,&H19,&H4E,0,0,&HF7}
 70 /*
 80 mode_change(sequencer)
 90 /*
100 end
110 func mode_change(a;char)
120 int fn
130     ex(5)=a
140     fn = fopen("midib","w")
150     fwrite(ex,8,fn)
160     fclose(fn)
170 endfunc
```

リスト2

```
 10 midi_init()
 20 bulk()
 30 end
 40 /*
 50 func midi_init()
 60   int fn
 70   fn=fopen("midi","w")
 80   fwrites("(RE)",fn)
 90   fclose(fn)
100 endfunc
110 /*
120 end
130 func bulk()
140 int fn
150 char a
160   fn=fopen("midib","r")
170   repeat
180      a=fgetc(fn)
190      switch a¥16 of
200         case 0
210         case 1
220         case 2
230         case 3
240         case 4
250         case 5
260         case 6
270         case 7:  color 3       :break
280         case 8:  color 2:print:break
290         case 9:  color 1
300         default :print
310      endswitch
320      print hex$(a);" ";
330   until inkey$(0)=chr$(27)
340   fclose(fn):color 3
350 endfunc
```

リスト3

```
 10 int in,in2,t1=0,q,x,y
 20 str kb$=" azsx cfvgb njmk,l. /:]",code,k$
 30 m_alloc(5,2000):m_assign(5,5)
 40 print:print"m_trk(5,";chr$(&H22);
 50 x=pos:y=csrlin
 60 repeat
 70   t1=t1+1
 80   k$=inkey$(0)
 90   if k$=" " then t1=0
100   if iscntrl(asc(k$)) then print chr$(asc(k$));  else {
110     in=instr(1,kb$,k$)
120     if in>=2  then {
130        q=pow(2,int(4-log(t1/10+1)/log(2)))
140      if code<>"" then {
150          print string$(len(code)/4-1,chr$(&H1D));
160          print itoa(q);string$(len(code)/4,chr$(&H1C));
170          print mid$("<>",(in2/14-((in2/7) mod 2)*2+1),1);
180            }
190      t1=0
200      code=encode(in)
210      in2=in/2+4
220      m_init():m_trk(5,code):m_play(5)
230      print code;
240     }
250   }
260 until k$=chr$(&H1B)
270 print chr$(&H22);")":print
280 print "m_play(5)":locate x,y-2
290 m_init()
300 end
310 func str encode(in)
320  return(mid$("><",((in/2+4)/14-(((in/2+4)/7) mod 2)*2+1),1
)+chr$((((in/2)-1) mod 7)+&H41)+left$("-",(in-1) mod 2))
330 end func
```

CP-40

効率よくしようと考えたものですが，ある程度楽器が弾けなければあまり使えません。

ところで，先日，楽器屋でちょっとしたオモチャをみつけました。RolandのCP-40というユニットで“Pitch to MIDI Converter”と書いてあります。要するにマイクから入力された音声をMIDI信号に変換して出力するのです。そういえば，PC-9801で「はなうたくん」というのがあったっけ……と思い出しました。てっきりPC-9801専用だと思っていたのですが，出力されるのは世界共通のMIDI信号です。プログラムは必要なら作ればすみます。

これは！　と思って購入しましたが，結論からいうと絶対音感がなければあまり使えません。ベンドや発音後の音量変化までサポートした機能充実ぶりはいいのですが，ちゃんと歌う（ハミングがいい）というのはちゃんと弾くのより難しいような気がします（昔合唱をやっていたこともありましたけど……)。ピアノ音などをセットしてテレビの音声を入れてやると前衛音楽が聞けますし，自分の音域が3オクターブと1音半だとわかったりしますので遊ぶにはいいかもしれません。

ま，とにかく，MIDIというとシンセサイザ関係が目立ちますが，なにも楽器だけとはかぎりません。そしてMIDIがサポートされているかぎり，どんなものでもX68000で扱えるわけです。

音長を知る

EOX(閑話休題)。それでは，いきなりですがMIDI INから送られてくる信号をリアルタイムに楽譜表示する……というところまではいきませんが，それに近いものを目指してみましょう。

ここまでで音階を知ることはできました。さらに発音時間を時間をカウントすることが必要です。これは音を出し終わるまで確定しない情報です。さらに残念ながらMIDIBからの信号はウエイトが入ってしまうので時間の計測が困難です。先ほどのプログラムのようにカウンタを回すことができないのです。試しにリアルタイムに信号の読めるmin( )関数を使ったところうまくいったのですが，インタプリタでは処理が重すぎてとてつもない誤差が出ます。ここではコンパイラ使用を前提にして，ファイル処理だけでできるだけなんとかしてみましょう。

いろいろ考えた末，ひとつのトラックでの和音を禁止することで解決しました。鍵盤をひとつずつ押さなければならないのは間抜けですが，ここまでの処理を見てわかるとおり，鍵盤がどのように押されてもプログラムで対応することは可能です。これはあくまで処理系の都合であり，技術的な障害はほとんどありません。

さて，ノートオンされてからノートオフするまでの時間はどのように計ったらよいでしょうか。ループ中でカウンタを回して時間を計ることは先ほどの理由でできません。また，最終的に楽譜表記にするならかなり大幅なクォンタイズ（時間情報を量子化すること）をかけてもかまわないでしょう（要するに分解能を粗くする)。

ここではコンパイルすることも考慮して，音楽ドライバのm_stat ( )，すなわち演奏中かどうかを調べる関数を使用しました。以前，「Yet Anather Column」で泉大介氏がコンパイルしても変わらないウエイトとしてこの機能を使ったこともありましたね。

プログラムを見るとわかるように，全音符から16分音符まで8段階の休符を同時に演奏してノートオフが入ったときにどこまで演奏中のトラックが残っているかで時間を計測しています。

ノートオフで鍵盤が離されたことは確実に検出できるのですが，ここでは1トラック1音の割り当てとしていますから，ある音がノートオンしたまま次の音のノートオンが入ってきた場合には前の音が消えたとみなしてやらねばなりません（楽器側で解決できる場合もある)。また，新しい音に切り替えていても前の音のノートオフ信号を拾ってキーオフしないように気をつけます。

これでリスト5の完成です。

実行してもいくつか数字が並ぶだけですが，画面の左端に音符情報（音階＋音長。音長は全音符を192としたときの値)の確定したもの，その下に現在演奏中の音階が表示されます。わかりやすくするため休符には対応していません。

もうひとつわかりやすさ優先のために手抜きした部分はクォンタイズの値です。ここでは演奏トラックに計りたい時間そのものを与えていますが，これでは4分音符に1ミリ秒間に合わなかっただけで付点8分音符になってしまいます。算数でいうところの切り捨てをやっているのですが，これはもちろん四捨五入風の処理が望ましい部分ですね。対応はきわめて簡単，休符の音長を変更すれば解決です。

ついでといってはなんですが，もうひとつ意外な手抜きがあります。それは音楽演奏関数です。なんで？　と思われる方もいるかもしれませんが，コンパイルを前提にしているからにはMUSIC.FNC用のライブラリを使うわけにもいきません。コンパイ

リスト4

```
 10  int fn,np,n
 20  int ch(16)
 30 /*
 40 cls
 50 midi_init()
 60 for i=0 to 15:ch(i)=0:next
 70 bulk()
 80 end
 90 /*
100 func midi_init()
110   int fn
120   fn=fopen("midi","w")
130   fwrites("(RE)",fn)
140   fclose(fn)
150 endfunc
160 /*
170 func bulk()
180 char a
190 int fn
200   fn=fopen("midib","r")
210   repeat
220       a=fgetc(fn)
240       if a¥16=8 then a=fgetc(fn):n=note_off(a)
250       if a¥16=9 then a=fgetc(fn):n=note_on(a)
260       locate 0,n
270       print hex$(ch(n));" "
280   until inkey$(0)=chr$(27)
290   fclose(fn)
300 endfunc
310 /*
320 func note_on(a)
330   int p=-1
340   repeat
350     p=p+1
360   until (ch(p)=0 or p>15)
370   ch(p)=a
380   return(p)
390 endfunc
400 func note_off(a)
410   int p=-1
420   repeat
430   p=p+1
440   until (ch(p)=a or p>15)
450     ch(p)=0
460   return(p)
470 endfunc
```

ルされたオブジェクトが内部でOPMDRVを呼び出しているからです。

m_alloc( )やm_play( )などたいていのものはファイル処理で代用できるのですが（1月号付録ディスクに収録されているKlondikeのソース参照），m_stat( )だけはどうにもなりません。MUSICDRV と OPMDRV を共存させて，OPMDRV を使って時間を計るしかないでしょう。それにつけてもMUSICDRV用のライブラリが待望されます。

アナログスティック？

シンセサイザにはときとしてジョイスティックがついていることがあります。これはピッチベンドやモジュレーション操作用につけられているものなのですが，これが実にアナログジョイスティックなわけです。というわけで安直ですが，BASICから楽器についているジョイスティックをアナログジョイスティックとして扱うプログラムを作ってみましょう。

このあたりの仕様は楽器によってマチマチなので，ここでは一応KORGのＭ１を想定して話を進めます。Ｍ１のスティック部分は横方向がベンド，上方向がピッチモジュレーション，下方向がVDFモジュレーションに割り当てられています。

これらのメッセージは３バイトの情報でやり取りされます。最初の１バイト（ステイタスバイト）は処理の種類，残りの２バイトを使ってデータを表しています。

ベンドは2ndバイトと3rdバイトをあわせた14ビットのデータです。左端が0000_H，中央が0020_H，右端が$7F7F_H$となっています。上下バイトを入れ替えればデータははっきりしてきます。

モジュレーションは2ndバイトでピッチモジュレーションかVDFモジュレーションかを判別し，3rdバイトで７ビット幅のデータを送っています。

こういった情報はマニュアルに付属しているインプリメンテーションチャートを見れば必ず載っていますので，他機種の方も確認してみてください。

ちなみに，ピッチベンド幅は楽器あるいは音色によって異なります。おおまかにいって，めいっぱい変化させたときの音程の変化が１オクターブになるもの，１音の幅になるものの２種類があります。手動で操作する場合は１音幅のほうが便利ですが，パソコンで扱う際は１オクターブ幅を標準としておいてよいでしょう。標準で１音幅の楽器もたいてい１オクターブ幅に設定し直すことができます。Ｍ１の場合はEDIT PROGRAMモードF7-2-Aの「ジョイスティックによるピッチ変化の最大値」を12に設定します。これは各音色ごとに設定されているのですべて書き換えておいたほうがよいかもしれません。

余談はともかく，リスト６がアナログスティック風プログラム。ジョイスティックのデータに従って画面に点を打ちます。あらかじめ断っておきますが，決して扱いやすいものではありませんし実用には向いていません。

さらに鍵盤をパソコンのキーボードに見立てて文字入力をし，そのリストを楽譜で……という計画もありましたが，あまりに不毛なのでやめておきます（ある種の人にはリスト入力が楽になるかもしれない）。

音と色

AMIGAやMacintoshのアプリケーションに音楽演奏をグラフィックとして表現するものがあります。演奏内容によってさまざまな模様が画面に広がっていくのです。X68000ではこのようなものはないようですので(私の知るかぎり)，ちょっとだけ検討してみましょう。

まず，BASICで描ける図形を見てみますと，X-BASICで可能なグラフィック関数のうち使いものになりそうなのはbox( )とcircle( )くらいです。速度的にはbox( )が有利なので今回はbox（ ）を使ってみま

リスト5

```
 10  int fn,np,n
 20  char a,b
 30  int ch,co,pch
 40 /*
 50 cls
 60 m_init()
 70 midi_init()
 80 for i=1 to 8
 90 m_alloc(i,16)
100 m_assign(i,i)
110 next
120 m_trk(1,"r16")
130 m_trk(2,"r8 ")
140 m_trk(3,"r8.")
150 m_trk(4,"r4 ")
160 m_trk(5,"r4.")
170 m_trk(6,"r2 ")
180 m_trk(7,"r2.")
190 m_trk(8,"r1 ")
200 ch=0
210 /*
220   fn=fopen("midib","r")
230   repeat
240      a=fgetc(fn)
250      switch a¥16 of
260         case 8:  b=fgetc(fn):note_off(b)  :break
270         case 9:  b=fgetc(fn):note_on(b)
280      endswitch
290   if m_stat(8)=0 then m_play():co=192          else {
300   if m_stat(7)=0 then m_play():co=192/2+192/4  else {
310   if m_stat(6)=0 then m_play():co=192/2          else {
320   if m_stat(5)=0 then m_play():co=192/4+192/8    else {
330   if m_stat(4)=0 then m_play():co=192/4          else {
340   if m_stat(3)=0 then m_play():co=192/8+192/16 else {
350   if m_stat(2)=0 then m_play():co=192/8          else {
360   if m_stat(1)=0 then m_play():co=192/16
370   }}}}}}}
380     locate 0,1
390     print hex$(ch);"  ";
400     locate 0,0
410     if co>0 then print hex$(pch);"  ";co;"  "
420   until inkey$(0)=chr$(27)
430   fclose(fn)
440 end
450 /*
460 func note_on(b)
470   if ch<>0 then pch=ch
480   if co<>-1 then note_off(b)
490   ch=b :co=0
500 endfunc
510 func note_off(b)
520   if b=ch then pch=ch:ch=0:l=co:co=-1
530 endfunc
540 /*
550 func midi_init()
560   int fn
570   fn=fopen("midi","w")
580   fwrites("(RE)",fn)
590   fclose(fn)
600 endfunc
```

リスト6

```
 10 int x=256,y=256,fn,fn2
 20 char a
 30 screen 1,3,1,1
 40 fn=fopen("midib","r")
 50 fn2=fopen("midi","w")
 60 fwrites("(RE)",fn2)
 70 fclose(fn2)
 80 repeat
 90   a=fgetc(fn)
100   if a¥16=&HB then {
110                if (fgetc(fn)=1) then {
120                                          y=fgetc(fn)*2+256
130                                        } else {
140                                          y=256-fgetc(fn)*2
150                                        }
160                        }
170   if a¥16=&HE then x=(fgetc(fn)+fgetc(fn)*&H7F)/32
180   pset(x,y,65535)
190   locate 0,0:print x,y
200 until inkey$(0)=chr$(27)
210 fclose(fn)
220 end
```

しょう。

問題はMIDI信号をいかに表示すべきかということです。簡単に得られる情報としては音階と音量がありますので，とりあえずこれをbox（ ）のX成分，Y成分としてみました。試しに表示してみても，まあ，妥当な解釈といえそうです。

残るは，「色」はどうやって選択すべきかというところです。まったく適当に出したのでは面白くありません。曲調から色彩を判断して……などができるほどの技術はないので，ここは先人の知恵にすがります。

調べてみると音楽と色彩を対応づけるような研究はいろいろ行われていることがわかります。残念ながら決定的な関係法則は見いだされていませんが，なんとか使えそうなものとして，キルヒャーの色彩音楽論による音程と色彩の対応表として知られるものをみつけました。表1がそれです。

音楽の時間に習ったことがあると思いますが音程とは2つの音階の音の関係で，図1に示されるようなものです。送られてくる音階から2音の音程を取れば，これを色彩に置き換えていくだけでよさそうです。

ぬか喜びしたのもつかのま，よーく見ると表1と図1では個数が違います。さらに，表1では増4度と減5度には別の色が割り当ててあります（これは聞く分にはおんなじ音程)。つらつらと考えてみると，どうもキルヒャーという人は平均律以前のお方のようです。純正律と平均律では音の響きや協和音の構成音が違ってきますが，ほかに適当なものもないのでこれを元にとにかくやってみることにしましょう。

さて，ドとミは長3度，ファとシは増4度，ドとファは完全4度……，与えられた2つの音からこういった関係を割り出すのは難しそうに見えます。しかし，音階で考えると面倒でも，図1を見てもわかるように，半音単位で考えれば並びは単純です。MIDIで送られてくるノート信号はまさに半音単位ですので，音程を知ることは実はぞうさありません。

それでできたプログラムがリスト7です。意外と単純。最新の32信号をbox（ ）で表示し，中心点は三角関数を組み合わせて滑らか，かつできるだけ不規則に移動するようにしています。

あとはMIDI信号をMIDI INから流し込んでやれば表示を開始します。

インタプリタ上での動作に限定するならfgetc (fn) の部分をmin( )にするとさらに快適な表示となります。

＊

さて，MUSICDRVを使ってX-BASICでさまざまな処理を行ってみました。私自身はMIDI楽器にいちばん近いイメージとしてプリンタを思い浮かべます。プリンタも遊ぶと結構面白いものですし，周辺機器制御という意味ではまったく同じ，扱い方もよく似ています。

周辺機器が高度になると，どうも与えられた使い方しかできない人が多くなりがちです。受動的でカタログスペックに流される傾向が強いとなにかを見失うような気もします。本来，用途を広げるための周辺機器によって，かえって使い方の幅が狭くなることさえありそうです。

MIDI特集でこういうのもなんですが，MIDIがなくとも，内蔵音源で十分に「音楽」することも可能です。「いくらがんばっててても内蔵音源じゃね」という言葉は50%の真実でしかありません。よくいわれるように，ハードの性能とユーザーの性能は掛け算しなければ答えは出ませんから。

フェラーリを駆るセーフティドライバー，アスファルトしか踏んだことのないモトクロッサー……悪いとはいいませんが，なにか違和感を感じます。私にとってX68000とは「あらゆる面で遊べる」マシンなのです。ユーザーが遊ぶこと，楽しむことこそ本質だと考えています。もっともっと遊び方を広げようではありませんか。

表1　音程と色との対応(ヴェレク，A., 1963)

	色	音程
オクターブ	白	短2度
	黄	短3度
	明（赤）	長3度
	金黄（黄橙）	5度
	（青）菫	短6度
	緑	8度
オクターブ	赤菫	7度
	青	減5度
	黒褐（濃灰）	増4度
	ばら（橙）	4度
	灰（明灰）	小全音
	黒	大全音

図1　度数

完全1度　短2度　長2度　短3度　長3度　完全4度　増4度　減5度　完全5度　増5度　短6度　長6度　短7度　長7度　完全8度

リスト7

```
 10 float t
 20 int fn,np,n,n2,n3,pc,xx,yy
 30 int x(31) ,y(31) ,cx(31),cy(31)
 40 /*
 50 cls
 60 screen 1,2,1,1
 70 vpage(3)
 80 palet(33,33333)
 90 midi_init()
100 demo()
110 width 96
120 end
130 /*
140 func midi_init()
150   int fn
160   fn=fopen("midi","w")
170   fwrites("(RE)",fn)
180   fclose(fn)
190 endfunc
200 /*
210 func demo()
220 char v
230 int fn,c
240   fn=fopen("midib","r")
250   repeat
260      a=fgetc(fn)
270      n=(n+1) mod 32
280      n2=(n+1) mod 32
290      if a¥16=9 then {
300             xx=fgetc(fn):yy=fgetc(fn)
310             n3=(n+31) mod 32
320             pc=(x(n3)-x(n)+12) mod 12
330             switch pc
340                  case  0: c=63420 :break
350                  case  1: c=65534 :break
360                  case  2: c=26487 :break
370                  case  3: c=61252 :break
380                  case  4: c=12234 :break
390                  case  5: c=42280 :break
400                  case  6: c=2688  :break
410                  case  7: c=64908 :break
420                  case  8: c=22282 :break
430                  case  9: c=510   :break
440                  case 10: c=8176  :break
450                  case 11: c=38514
460             endswitch
470             }
480      x(n)=xx:y(n)=yy
490      t=t+0.01#
500      cx(n)=sin(t*4.99#)*cos(t/2.04#)*128+256
510      cy(n)=cos(t*3#)*cos(t*7#)*128+256
520      palet(n+1,c)
530      apage(1)
540      box(cx(n)-x(n),cy(n)-y(n),cx(n)+x(n),cy(n)+y(n),n+1)
550      box(cx(n2)-x(n2),cy(n2)-y(n2),cx(n2)+x(n2),cy(n2)+y(n2
),0)
560      apage(0)
570      pset(cx(n),cy(n),33)
580   until inkey$(0)=chr$(27)
590   fclose(fn)
600 endfunc
```

MUSICDRV用演奏データ

ヴァルナより町のテーマ

ⒸSYSTEM SACOM

Nishikawa Zenji 西川 善司

MUSICDRVを使ったサンプルプログラムです。手軽で高性能な音源として注目されるRolandのU220に対応しています。その他の機種への変更も比較的簡単ですので手持ちの楽器で聞いてみてください。

突然,MIDI特集ということをいわれたのでなんの準備もしていなかった私は今回あわてて作った1曲を発表して逃げることにします。曲はシステムサコムから出ていたPC-8801用のゲーム「ヴァルナ」の町のテーマです。作曲はX1ユーザーなら誰でもご存じの「ユーフォリー」の作曲者である斎藤学氏です。「ヴァルナ」はイースタイプのRPGで……そう,確かこのゲームが発表された年の1989年はまさにARPGの年でした。サコムは「ヴァルナ」のほかにも「プロヴィデンス」,テクノソフトは「新九玉伝」,T&Eは「ハイドライド3」,エニックスは「オールドビレッジストーリー」,ファルコムは「イースIII」と各メーカーこぞってARPGを出しあってたものでしたね。話が逸れましたが,この「ヴァルナ」,1曲1曲は「ユーフォリー」同様短いのですが一度聞いたら思わず口ずさめてしまうような美しいメロディのオンパレード! で,私はテープに録って幾度となく聞いたものでした(しみじみ)。

プログラムは譜面からでなくて聞き取りのうえ時間もなかったのでアレンジは結構いい加減です。

入力と実行

一応対応楽器はローランドU220ということにしておきますがとても短い曲で,しかも使用した音色は機種に依存したものではないので移植は非常に楽かと思われます(リスト後部に使用音色名を記しておきます)。

また,演奏に使用している各音色のピッチベンドの範囲は各自-12,+12に修正しておいてください(U220マニュアル68ページ参照。工場出荷状態では-2,+2となっています)。他機種に移植する方も同様ですが,MT-32に限っては電源投入時にそう設定されてしまいますので特に気にする必要はありません。

なお,パッチはMIDIチャンネルの1～6とリズムチャンネルは10に設定しておいてください。また,ボリューム,パンポット,ホールドなどのMIDI信号も受信できるようにしておいてください(U220マニュアル59ページ参照)。

演奏にはMUSICDRV.XとMUSIC1.FNCが必要です。以上の2つはOh!X 1月号の付録ディスクに収められています。

ちょっと,おまけ

ここで少しU220の紹介をしておきましょうか。

皆さんご存じ「サンプラ」は生の音からサンプリングできるので音色の作成は楽なうえにリアル。当たり前ですね。しかし,サンプラはほとんどプリセット音色というものがない(と考えてよい)ので実際買ってきても路頭に迷ってしまう人が多いのです。U220はそんな,音質にこだわりたいちょっぴり「おませさん」向きの楽器なのです。

音は全音サンプリングされたもので同時発声数も30と贅沢。もちろんマルチティンバーでリズムセットもついていますし,リバーブ,コーラスのエフェクタも内蔵。プリセット音色もかなり充実しているので買ってきたその日から遊べます。デモ曲の鳴らし方を書いておきますので楽器屋に寄ったときにでも聞いてみましょう。

1) まず「JUMP」キーを押しながらVALUEの「▲」を押す
2) CURSORの「◀」「▶」で曲を選択
3) 「ENTER」で演奏開始

また,このU220と同じ価格ラインでしかも性能的にも同レベルのものがあるのでそれらも挙げておきます。

U220	ROLAND	110,000円
M3R	KORG	124,000円
TG55	YAMAHA	110,000円
K-4r	KAWAI	99,800円

どの機種も音色の追加はカードで行えるわけですが,そのソフトの充実度はやはりU220がダントツのような気がします。また一度に2枚の音色カードが差さるのも

西川善司のデモテーププレゼント

私は去年の11月に某大学の学園祭で同大学のサークルと共同でX68000によるMIDI楽器の実演を行ったのでした。この大学,大都心東京新宿にありながら昨年の学園祭はてんで盛り上がらず,私たちのところには3日間でたった80人の来客しかありませんでした。とほほ。当日は気合いを入れて自分ちの機材全部を小型トラックに積んで運んだのですが,時期が時期。そう,あの大嘗祭の警戒体制のなか,うさんくさい積み荷満載で新宿まで走ったのです。あー,よく呼び止められなかったな。

当日の内容はというとシャープの未・体・験フェアみたいな感じでMIDIの説明とその実演でした。上にも書いたように本当に来客が少なく,喫茶店を大教室で開いていたある参加団体は,閑古鳥に絶えかねて室内野球をやっとりました(悲しい)。

で,このときの演奏をカセットテープにまとめてみました(ライブじゃないよ)。曲目は秘密。使用楽器のWAVESTATION, M1, T3, D10, U220, S330, R8M, RX-8, MT-32, M3Rなどを一括して24チャンネルミキサを通してノーダビングで録音しました(そうそう,なつかしのボコーダSVC-350も演奏に参加していますよ。そういえば一時期幻と化したボコーダですが,最近は中古市場で比較的容易に手に入るようで……)。シーケンスはほとんどがMUSICDRV.Xによるものです(一部OPMD)。マスターテープはカセットテープ(METAL)ですので音質はそこそこですが,とりあえずMIDIをやっている人にはなんらかの参考にはなると自負しております。また,上に列記した楽器のうち購入予定ものがあるとすれば,ちょっとしたガイドにもなることでしょう。

今回のテープ制作で活躍したのは,なんといってもKORGのWAVESTATION。これは実に音の作りやすいシンセで396種類の波形を最大32個もアサインできるのでFM音源っぽい音なら,ものの数分で作れてしまいます。まさにシンセマニア向けのシンセといった感じです(その半面プリセット音がイマイチ使えないけど)。

まあ,そんなわけでほしい人はアンケートはがきのプレゼント番号には「0」と書いてきてください。

U220だけ，おいしいよね。

U220の欠点といえば，

1) ボイスリザーブをしっかりやらないと発音遅れを生じる（MT-32もパーシャルリザーブをしっかりやらないと発音が遅れてたよねぇ，それとおんなじよ)。

2) サンプリングレートやビット数が怪しい。マニュアルにそのこと関係が全然書いていない（まあ，音はいいのでどうでもいいことかもしれないけど……。誰か知ってたら教えてほしいよ。多分12ビットの24kHzあたりだと思うんだけど)。

＊

人によっては「UがあればMはいらない」となにやら格言めいたことをいいだす人まで出てきたので，その真偽はあなたの耳で確かめてやってください（ちなみに上は「U220があればM1はいらない」の意)。

編注）このプログラムはU220用ですが，音色番号を変えれば他機種でもそれなりの演奏を聞くことができます（8チャンネルのマシンではリズムが出ない)。以下に手元にある楽器のものについて，代替音色例（リスト末尾の音色順のプログラム番号）を示しておきます。

●KORG　M1/R, T1/2/3
@32, @52, @05, @93, @19, @03, @07, @02
ただし，リズムなし。

●Roland　MT-32, CM-32L/64
@17, @06, @60, @25, @73, @96, @71, @01
リズムキットの変更は不要。

●Roland D10/20
@17, @06, @101, @38, @49, @37, @95, @01
リズムキットの変更は不要？

●YAMAHA　QY-10
@04, @02, @13, @22, @28, @09, @14, @01
ただし，リズムなし。

リスト1

```
 1050 /* MEMO   使用楽器 U220
 1060 /*
 1070 /*
 1080 m_tempo(113)
 1090 /*
 1100 key 2,"m_play()"+chr$(13):key 12,"m_stop()"+chr$(13):key 1
9,"save"+chr$(34)+"VATW"+chr$(5):key 7,"m_trk(":key 8,"m_solo()"
+chr$(&H1D)+chr$(1)
 1110 /*
 1120 /*
 1130 for i=1 to 20:m_alloc(i,5000):m_assign(i,i):next
 1140 /*
 1150 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
],aa[256]
 1160 str j[256],k[256],l[256],m[256],n[256],o[256],p[256],q[256
],r[256]
 1170 str s[256],t[256],u[256],w[256],x[256],y[256],z[256]
 1180 str a1[256],b1[256],c1[256],d1[256],e1[256],t1[256],g1[256
],h1[256]
 1190 str a2[256],b2[256],c2[256],d2[256],e2[256],f2[256],g2[256
],h2[256]
 1200 str j1[256],k1[256],l1[256],m1[256],n1[256],o1[256],p1[256
],q1[256],r1[256]
 1210 str j2[256],k2[256],l2[256],m2[256],n2[256],o2[256],p2[256
],q2[256],r2[256]
 1220 str s1[256],u1[256],v1[256],w1[256],x1[256],y1[256],z1[256
]
 1230 str s2[256],u2[256],v2[256],w2[256],x2[256],y2[256],z2[256
]
 1240 str dd[256],ee[256],ff[256],gg[256],h3[256],kk[256],jj[256
],p3[256]
 1250 /*
 1260 /*         TRACK 1 (MELODY)
 1270 /*
 1280 a="L16c+d
 1290 b="L16e2rc+rdrera g+4.f+8e2 f+8.d8.c+8>b8f+8a8b8< c+4rdrc+
>b4.<c+d
 1300 c="L16e2rc+rdrera g+4.{f+g+f+}8e2 f+8>b8<d8f+8a8g+8a8ba& a
&v12a&v11a&v10a&v9a&v8a&v7a&v5ar4@06r4
 1310 d="L16c+4f+4e4c+>b<rc+& c+1 c+4f+4g+4rarc+& c+1 d8e8f+8da8
.g+8f+8g+8 e8f+8g+8eb8.a8g+8a8 a8b8<c+8>a8b8<c+8>a8<c+8> @u110a2
g+4.@57r8
 1500 m_trk(1,"@n1 @57 o5 q8 v14 @u110@b8192 @m0 @p63"+a)
 1510 m_trk(1,"[do]"+b)
 1520 m_trk(1,c)
 1530 m_trk(1,"         o5 q8 v14 @u99"+d)
 1540 m_trk(1,"         o5 q8     @u110 [loop]")
 2000 /*
 2050 c="L16e2rc+rdrera g+4.{f+g+f+}8e2 f+8>b8<d8f+8a8g+8a8ba& a
&a&a&v3a&a&v2a&a&v1ar4@06r8r32
 2060 d="L16c+4f+4e4c+>b<rc+& c+1 c+4f+4g+4rarc+& c+1 @b8192v10>
b8<c+8d8>b<f+8.e8d8e8 c+8d8e8c+g+8.f+8e8f+8 f+8g+8a8f+8g+8a8f+8a
8 f+2e4..@57r16
 2500 m_trk(2,"@n2 @57 o5 q8 v4  @u110@b8292 @m0 @p63 r16."+a)
 2510 m_trk(2,"[do]"+b)
 2520 m_trk(2,c)
 2530 m_trk(2,"         o5 q8 v4  @u99"+d)
 2540 m_trk(2,"         o5 q8 v4  @u110 @b8292r16.[loop]")
 6000 /*
 6010 /*         TRACK 6 (BACK SYNTHE)
 6020 /*
 6030 a="r8
 6040 b="L16c+1 e2..ee- d2f+2 f+2g+2
 6050 c="L16c+1 e2..ee- d2'd2>b<' 'c+2>a<'r4@71r4
 6060 d="a1  e2e2  f+1 e2L32@d127{eg+c+&}8<c+4.@d0> d2g+2 e2a2 a
1 >a2g+4&g+16@72@b8192@p30r8.
 6500 m_trk(6,"@n3 @72 o5 q8 v10 @u99 @b8192 @m0 @p30"+a)
 6510 m_trk(6,"[do]"+b)
 6520 m_trk(6,c)
 6530 m_trk(6,"         o5 q8 v10 @u40 @b8292 @p0"+d)
 6540 m_trk(6,"         o5 q8     @u99 [loop]")
 7000 /*
 7010 /*         TRACK 7 (BACK FLUTE)
 7020 /*
 7030 a="r8
 7040 b="c+1 e1 d2f+2 f+2g+2
 7050 c="L16c+1 e1 d2d2 c+4&v7c+&v6c+&v4c+&v3c+r4.@74r8
 7060 d="c+1 e2c+2 c+1 c+2<c+2>                         d2g+2 e2a2 a
1 f+2e4@49@b8192@p90r4
 7500 m_trk(7,"@n4 @49 o5 q8 v10 @u60 @b8192 @m0 @p90"+a)
 7510 m_trk(7,"[do]"+b)
 7520 m_trk(7,c)
 7530 m_trk(7,"         o4 q8 v10 @u60 @b8292 @p127"+d)
 7540 m_trk(7,"         o5 q8 [loop]")
 8000 /*
 8010 /*         TRACK 8 (CHORD)
 8020 /*
 8030 a="r8
 8040 b="'a1<c+e>' '<c+1eg+>' 'b1<df+>' 'b2<df+>''b2<eg+>'
 8050 c="'a1<c+e>' '<c+1eg+>' 'b2<df+>''b2<eg+' @d127L16'c+4ea>'
v7rv6rv5rv4r@d0r4..@01r16
 8060 d="'>a1<c+f+' 'c+2eg+''>g+2<c+e'  '>a1<c+f+' 'c+2eg+''e2g+
<c+>'  '>b2<df+''>b2<eg+' 'c+2eg+''c+2f+a' 'c+1f+a' '>a2<df+''>g
+4...b<e'@82r32
 8500 m_trk(8,"@n5 @82 o4 q8 v10 @u70 @b8192 @m0 @p63"+a)
/*@21 でも good!!
 8510 m_trk(8,"[do]"+b)
 8520 m_trk(8,c)
 8530 m_trk(8,"         o4 q8 v10 @u70"+d)
 8540 m_trk(8,"o4[loop]")
 9000 /*
 9010 /*         TRACK 9 (BASS)
 9020 /*
 9030 a="r8
 9040 b="L16a8.a<a8>a4.<a8>a8< c+8.<c+8.>c+8c+4<c+8>c+8> b8.<b8.
>b4<b8>b8b8< e8.<e8.>e8e4e8>f+g
 9050 c="L16a8.<a8.>a4.<a8>a8< c+8.<c+8.>c+8c+4<c+8>c+8> b2<e2>
a8.<a8.>a8aaaar4
 9060 d="<f+1 c+1 f+1 c+1 >b2<e2 c+2f+2> b1 b2<e4>{b<cc+d>}4
 9500 m_trk(9,"@n6 @34 o2 q8 v10 @u90 @b8192 @m0 @p63"+a)
 9510 m_trk(9,"[do]"+b)
 9520 m_trk(9,c)
 9530 m_trk(9,d+"[loop]")
10000 /*
10010 /*         TRACK 10 (RYTHM)
10020 /*BD
10030 a="r8
10040 b="L4o2@u60|:4 crcr :|
10050 c="L4o2@u60|:crcr :| c2c2 cr2.
10060 d="L4o2@u60|:4 c1 :| cccc cccc cr2. cccr
10500 m_trk(10,"@n10 v15"+a)
10510 m_trk(10,"[do]"+b)
10520 m_trk(10,c)
10530 m_trk(10,d+"[loop]")
11000 /*SD
11030 a="r8
11040 b="L4o6@u90|:4 rdrd :|
11050 c="L4o6@u90|:rdrd :| d2d2 rd{dddd}o3@u80{rc>af}
11060 d="L4o3@u80|:7 r1 :| r2.{c>afr}
11500 m_trk(11,"@n10"+a)
11510 m_trk(11,"[do]"+b)
11520 m_trk(11,c)
11530 m_trk(11,d+"[loop]")
12000 /*HH
12030 a="r8
12040 b="L16o2|:4 @u40f+@u30f+f+f+f+f+f+f+f+f+f+f+f+f+f+ :|
12050 c="L16o2|: @u40f+@u30f+f+f+f+f+f+f+f+f+f+f+f+f+f+ :| |:@
u80<c+4>@u30f+f+f+f+:| |:@u40f+@u30f+f+f+f+f+f+a+f+a+f+f+a+f+f
+
12060 d="L16o2|:7 @u40f+8@u30f+f+ :|@u40f+@u30f+a+8":d=d+d+d+"L4
@u40f+f+f+f+ f+f+f+r
12500 m_trk(12,"@n10"+a)
12510 m_trk(12,"[do]"+b)
12520 m_trk(12,c)
12530 m_trk(12,d+"[loop]")
13000 /*TAMBARINE/CYMBAL
13030 a="r8
13040 b="L4o3@u80|:4 f+f+f+f+ :|
13050 c="L4o3@u80 |: f+f+f+f+ :| f+2f+2 r2{f+f+f+f+}r
13060 d="L4|:4 r1 :| o3@u80bbbb bbbb c+1 bbbr
13500 m_trk(13,"@n10"+a)
13510 m_trk(13,"[do]"+b)
13520 m_trk(13,c)
13530 m_trk(13,d+"[loop]")
19999 m_play():end
20000 /* @57 ﾊｰﾌﾟｼｺｰﾄﾞ/ｸﾗﾋﾞ系  @06 ｴﾚﾋﾟ  @71 ｱｺｰｽﾃｨｯｸｷﾞﾀｰ系  @72
 ｼﾝｾﾌﾞﾗｽ系
20010 /* @49 ﾌﾙｰﾄ  @82 ﾌﾞﾗｽ系  @34 ﾌﾚｯﾄﾚｽﾍﾞｰｽ  @01 ｱｺｰｽﾃｨｯｸﾋﾟｱﾉ
```

X68000用
戦いの兜
Yabe Masatoshi
矢部 雅敏

X1/turbo用
LITTLE WING
Abe Toshimitu
阿部 俊光

MIDI X68000用（要MT-32/CM-32/CM-64，MUSICDRV）
リゾ・ラバ(Resort Lovers)
Hazama Manabu
狭間 学

MIDI X68000用（要MT-32/CM-32L/CM-64，MUSICDRV）
花
Nakanishi Michikazu
中西 道一

今月は特集がMIDIということもあって，MIDI対応の曲も取り上げてみました。オリジナルあり，バンドあり，はたまた学校唱歌ありと多彩な顔ぶれ。ぜひ打ち込んで聴いてみてください。なお，予告していたビートルズの曲は，事情により掲載できなくなってしまいました。楽しみにしていた方，ごめんなさい。

久びさのオリジナルソング

世の中には作曲をしている人がいます。それでご飯を食べている人たちが作曲家と呼ばれているのは皆さんご存じでしょう。一般的な感覚では作曲をする人というと，ちょっと特別な才能の持ち主というような感じを受けます。しかし，趣味で絵を描く人がいるように趣味で曲を作る人がいることも事実なのです。今回登場していただいた矢部君もそのひとり。タイトルは「戦いの兜」です。戦国時代を意識して作ったとのこと。4パートで構成されています。

1：出陣
2：敵軍との間合いがつまっていく
3：大将同士の1対1の戦い
4：戦いから城に戻り，夜の星空を眺める

といったテーマが各パートに与えられています。

私の個人的な感想なのですが，全体的にメリハリが欠けているような気がします。動と静の差が少ないのではないでしょうか。その分，サビの盛り上がりが薄れていると思います。曲全体が短いこともあり，1つのパートが短めなのはしょうがないとしても，パート1とパート2で1部，パート3，4で1部という2部構成に聞こえてしまいます。パート2と4にもう少し工夫の余地が残っているのではないでしょうか。パート2ではだんだんコードのテンションを高めていくとか，パート4では思い切ってスローテンポなものにしてみるなど，テーマにそって，動と静を形作ってみてはどうでしょう。文章を書くときに「起承転結」などといいますが，音楽の進行にもあてはまる方法だと思います。全体的には戦闘的な雰囲気をよく表しており，「戦いの兜」というタイトルにあったものでしょう。矢部君はすでにひとつの曲を作り上げる力量を持っていると思われます。感性を磨いて精進してください。（S.K.）

小さな翼でもパワーは全開

きたきた，きました，LINDBERG。ヤングジャンプでアイドルとしてデビューしたけど，先のことを考えてか，最近の流行りに乗じてうまくバンドに転向したいい例です。ラ・ムーは失敗しちゃったもんねぇ。

さて，曲は「LITTLE WING」です。LINDBERGというと，ドラマ「世界で一番君が好き」で使われていた「今すぐKISS ME」がよく知られていますけど，実は隠れた名曲（でもないか）としてこの曲は人気なのです。最近はCMソングにも起用されているので，聴いたことのある人も多いと思います。

全体としてはまとまりもよく，安定したカンジがしてGOODです。ただ欲をいえば，ボーカルのメロディに，もちょっとメリハリというか元気さがほしかった。このバンドの場合，ボーカルの渡瀬マキのポップな元気さがウリだったりするので，弾むような音のキレが表現できるともっといいのに，なんて，ちょっとワガママかな。

最近ちょっと下火になったとはいえ，まだまだ巷はバンドブーム。そのせいか，このところこのLIVE inにもいくつかそのテの曲が届いています。いい傾向です。いろんな音楽を聴いて，いろんな音やリズムをどんどん吸収してほしいですね。ただ，やっぱりコンピュータミュージックの場合，どうしてもネックになるのがボーカル。人間の肉声＋その人のパワー，うまく表現するのはやっぱり難しいよね。だからこそ音選びは慎重にやってほしいです。バックはよくできているのに，ってボツになる場合も結構あるんです。大変だろうけど，ぜひぜひ頑張ってクリアしてください。（香）

LINDBERG

ぜんぶう〜そさっ，んっ!?

こんにちは，浦川です。実は私はD-20のユーザーだったのであります。最近ではLA音源に飽き足りなくなって，さらにW-30まで買ってしまい置き場に困っております。こんなことなら最初からCM-64を買っておけばよかった。しくしく。そういう苦い経験を買われて（？）今月はMIDIバージョンのほうのコメンテーターをさせてもらうことになりましたのでよろしく。

最初の曲はBAKUFU-SLUMPの「リゾ・ラバ（Resort Lovers）」。コスモ石油の夏のCMソングになっていました。「ぜんぶうーそさっ，そんなもーんさっ」と歌いながら町を走る小学生をあなたも見たことがあるでしょう。私は3回あるぞ。Aメロ，Bメロとひと夏の恋を歌い上げておいて，サビになると「そんなわきゃねーよっ！」と豹変するという，いかにもBAKUFU-

BAKUFU-SLUMP

SLUMPらしい（？）曲です。

作ってくれたのは，以前はMZでも頑張っていた狭間学くん。さすがにX68000とMT-32との音色の割り振りも手慣れたもんです。ギターのカッティングもていねいに音を取っているのは偉いですね。ブラスの選択もいいし，ボーカルを前面に出したミキシングも成功してるし，全体に非常に洗練された仕上がりを見せています。BAKUFU-SLUMPはアクの強い演奏をするメンバーばかりなので，コンピュータミュージックにするとまた違った表情が出て面白いですね。

今後の課題としては，メロディをありがちな音色に頼ってしまっている点です。AtmosphereとSchooldazeという音色はLA音源にしては素直な音なのでつい使ってしまいますが，「ミュージ君」のサンプル曲でもメロディがたいていこの系統。悪いというわけでもありませんが，サンプル曲と間違えられるのも損な話。サンプラザ中野のボーカルがキーですから，もっと太い音色と組み合わせてみたら面白かったと思います。あと，エレピがややなおざりなのも気になるところです。

ああ，懐かしの学校唱歌

続いての曲は滝廉太郎の「花」。「春のうらぁらぁの，すーみーだーがーわぁ～」というアレですね。音楽の時間なんかによく歌わされた曲。私は最近までこの曲のタイトルを「すみだ川」だと思ってました。

この「花」という曲は組歌「四季」の一部だそうです。「四季」は「花」「納涼」「月」「雪」の4つから成っていて，こういった洋楽のスタイルの曲を作ったのは，彼が初めてだったんですね。滝廉太郎は一方では純日本風の曲「荒城の月」も発表したりして，かなり多彩な才能を持った人でした。

送ってくれた中西道一君はSplash waveのおまけのつもりだったようですが，こちらが採用になってしまいました。まぁ，人生いろいろってヤツですな。

Choraleとピアノだけというシンプルな構成ですが，音色がいいのでかえってイヤミのないまとまりを見せています。ピアノを中央におき，2つのパートが左右に分かれて「ふふふー」と歌うのも音楽室っぽくてなかなか。ただ，ピアノがちょっと投げやりなのが難点。今のままだと，テンポと大きさに大きな変化がないので投げやりに聞こえてしまいます。D-20のシーケンサを使っている人は自分が上手く弾く以外に表情をつける方法はないけど，MMLならそのへんの調整はラクにできるはずだから，もう少しがんばってほしかったですね。

歌曲は，構成はシンプルなかわりに各パートの表情には神経を使うジャンルです。練習用に向いているので，初心者の方は打ち込んでいろいろいじってみると勉強になると思いますよ。 (浦)

リスト1　戦いの兜

```
  10 /* key 3,"m_trk("
  20 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  30 /*                                  /*
  40 /*  オリジナル  MUSIC              /*
  50 /*               戦い の 兜         /*
  60 /*                 ORIGINAL NO.36   /*
  70 /*  作曲  矢部雅敏 （  一般素人  ）  /*
  80 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  90 m_init()
 100 /*
 110 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
],j[256]
 120 str k[256],l[256],m[256],n[256],o[256],p[256],q[256]
 130 str c1[256],c2[256],c3[256],d1[256]
 140 dim char v(4,10)
 150 /*      B A S S
 160 v={
 170 43,15,0,0,0,0,0,0,0,3,0,
 180 31,16,15,0,5,20,2,10,-3,0,0,
 190 20,10,8,0,5,43,2,1,3,0,0,
 200 28,4,8,0,11,23,1,0,0,0,0,
 210 31,4,8,6,11,0,1,0,0,0,0}
 220 m_vset(70,v)
 230 /* G U I T E R
 240 v={19,15,0,0,0,0,0,0,0,3,0,
 250 31,0,0,6,0,20,0,5,0,0,0,
 260 22,11,0,7,1,47,0,13,2,0,0,
 270 18,0,0,7,0,18,0,0,7,0,0,
 280 31,8,0,7,1,1,0,0,3,0,0}
 290 m_vset(78,v)
 300 /*    B    E    L    L
 310 v={4,15,0,0,0,0,0,0,0,3,0,
 320 31,12,0,6,2,39,0,10,3,0,0,
 330 31,8,0,8,15,2,0,2,3,0,0,
 340 31,12,8,6,15,44,0,7,4,0,0,
 350 31,8,0,8,15,2,0,1,4,0,0}
 360 m_vset(76,v)
 370 /*/* H A R D   B R A S S
 380 v={56,15,0,0,0,0,0,0,0,3,0,
 390 17,16,0,3,3,26,0,3,3,0,0,
 400 20,0,0,3,0,27,0,1,0,0,0,
 410 28,0,0,3,0,28,0,1,0,0,0,
 420 28,0,0,8,0,0,0,1,4,0,0}
 430 m_vset(79,v)
 440 /*     D・G U I T E R
 450 v={40,15,3,0,200,127,0,0,0,3,0,
 460 31,2,1,0,0,8,0,3,0,0,0,
 470 18,1,1,8,0,22,0,15,0,0,0,
 480 31,2,1,0,0,25,0,7,1,0,0,
 490 31,14,0,8,0,3,0,1,7,0,0}
 500 m_vset(81,v)
 510 /* H I   H A T
 520 v={59,15,0,0,0,0,0,0,0,3,0,
 530 31,31,0,3,0,0,0,5,0,1,0,
 540 31,23,0,5,10,20,0,0,0,0,0,
 550 31,22,12,8,15,20,0,2,0,1,0,
 560 31,0,12,7,0,0,2,2,0,0,0}
 570 m_vset(95,v)
 580 /*/* D. G U I T E R  2 ( N. A R M S )
 590 v={56,15,2,0,210,40,0,1,0,3,0,
 600 28,8,0,15,12,24,2,3,7,0,0,
 610 30,4,0,15,12,18,0,1,0,0,0,
 620 28,0,0,15,15,20,0,1,0,0,0,
 630 30,4,5,15,15,0,0,1,0,0,0}
 640 m_vset(101,v)
 650 for i=1 to 8:m_alloc(i,8000):m_assign(i,i):next
 660 /* S U B M E L O D Y   G U I T E R  1
 670 a="t170 l8r8r1 @78 @v125
 680 b="o4  r2 c2d2  r1 f2g2 r2
 690 c="@v122r4g1 f2  cd4.f1rrr2
 700 d="r4@v127f2r2g2r2c4d2g1
 710 e="r4d1  d4.e4.a1g1
 720 f="rf4.a1g2<c1d1
 730 g=">g2a4 d4d4ef2 e4g1 r4rr2
 740 h="ggag4 r4r r1r1r1
 750 k="ffgf4 r4r r1r1r1
 760 n="ffff4raaaa4o4rrr
 770 o="@v127r4 e2f4g4 a2  gfe4 c2 >b<c4d2r4r
 780 p=">g4a4b4<c2  >a4b<c4dc2 d4de2rr2
 790 q=">f4g4 ab<c2 d4 cf4<c1r>r  rrrr4r4 rrr f4f2
 800 m_trk(1,a)/*/*/*良い子のためのだじゃれ講座
 810 m_trk(1,b)/*隣の囲いに外人が来たんだって！ へい、かもん。
 820 m_trk(1,c)/*/*/*/*/*/*/*
 830 m_trk(1,b)/*ナイアスが、売って ナイアス。
 840 m_trk(1,c)/*幻獣鬼 の  ボス  には、厳重に気をつけよう！！
 850 m_trk(1,d)/*A君「ビデオを  はがす時、痛むんですよ！」
 860 m_trk(1,e)/*B君「それは、サビオだろ！」
 870 m_trk(1,f)/*「僕ちゃん今日、かぜ引いて、だるいあす。」
 880 m_trk(1,g)/*「あちしも今日、かぜ引いて、だるいあす2。」
 890 m_trk(1,h)/*「オパオーパ」B君「ウーハールーハーだろ！」
 900 m_trk(1,k)/*とど捕りはつらいな！」B君「音採り  だろ！」
 910 m_trk(1,h)/*/*/*/*メッセージ
 920 m_trk(1,k)/*誰か、ラグーン幻狭の丘  の音楽 を、
 930 m_trk(1,n)/*作ってOh！Xに、出してください。
 940 m_trk(1,o)/*闇の血族  上巻の  ENDING も
 950 m_trk(1,p)/*      誰か、作って、欲しいのだだ！
 960 m_trk(1,o):m_trk(1,q)
 970 /* S U B M E L O D Y   G U I T E R  2
 980 a="t170 l8r8 r1 @78 @v125 y49,64
 990 m_trk(2,a):m_trk(2,b)
1000 m_trk(2,c):m_trk(2,b)
1010 m_trk(2,c):m_trk(2,d)
1020 m_trk(2,e):m_trk(2,f)
1030 m_trk(2,g):m_trk(2,h)
1040 m_trk(2,k):m_trk(2,h)
1050 m_trk(2,k):m_trk(2,n)
1060 m_trk(2,o):m_trk(2,p)
1070 m_trk(2,o):m_trk(2,q)
1080 /*  G R A S S   B E L L
```

```
1090 a=" o3 @v122 18 @76
1100 b="|:4dd<d>d:||:4cc<c>c:|
1110 c="|:4d-d-<d->d-:||:4e-e-<e->e-:|
1120 d="dddd4rdddd4r8r8r8
1130 e="@v122|:2rrd4:||:2rre4:|
1140 m_trk(3,a+"r8r1")
1150 m_trk(3,b+b+b+b)
1160 m_trk(3,b+b+b+b)
1170 m_trk(3,b+b+b+b)
1180 m_trk(3,d)
1190 m_trk(3,o+p+o)
1200 m_trk(3,q)
1210 /*/* B A C K I N G   D I S T O S I O N . G  1
1220 a="t170 o3  @v118 18 @101"
1230 b="g2 g2g1  d2 f2g1
1240 c="c4 cd4g2 gfdcdc>a <d4c2 @v122@81f16e16d16c16 g1@101@v11
8
1250 d="c4 cd4g2 gfdcdc>a <d4fg2 @v122@81f4edc4 r@v118>r4
1260 e="<@101a1fede2gb>ff<f4>f<f2&f8b-2r
1270 f="c2d2 f4  g2 cde gacd4a2rrr2
1280 g="agf a2 def a2g4 fede g4.<c1r8
1290 h=">a2gf2 def g4<dg1r r1
1300 j="ggag4@v126@81 >d2efe<d2cdf2>ffec2r8r4
1310 k="@v122ffgf4@v126 <a2 g16f16e16d16f16e16d16c16 >a16g16f16
e16dg4.<c1r2r
1320 l="@v122<ggag4>@v126> a2 gfe d4e4 fg16a2 gf16e16d16 e2 r2
<
1330 m="@v122<ffgf4>@v126> a2ab<c4 dc4 e4f a2g1r
1340 n="ffff4raaaa4o3@v115@101r8r8r8
1350 o="ffff ffff ff<f>f ff<f>f  aaaa aaaa aa<a>a aa<a>a
1360 p="gggg gggg gg<g>g gg<g>g<cccc cccc cc<c>c cc<c>c>
1370 q=">gggg gggg gggg gggg <cccc cccc @81@v125 aaa a4g4rrr f4
f2
1380 m_trk(4,a+"r8r1")
1390 m_trk(4,b):m_trk(4,c)
1400 m_trk(4,b):m_trk(4,d)
1410 m_trk(4,e):m_trk(4,f):m_trk(4,g):m_trk(4,h)
1420 m_trk(4,j):m_trk(4,k):m_trk(4,l):m_trk(4,m)
1430 m_trk(4,n):m_trk(4,o):m_trk(4,p):m_trk(4,o)
1440 m_trk(4,q)
1450 /*/* B A C K I N G   D . G U I T E R  ( D E T )
1460 a="t170 o3  @v118 18 @101y52,32"
1470 m_trk(5,a+"r8r1")
1480 m_trk(5,b):m_trk(5,c)
1490 m_trk(5,b):m_trk(5,d)
1500 m_trk(5,e):m_trk(5,f):m_trk(5,g):m_trk(5,h)
1510 m_trk(5,j):m_trk(5,k):m_trk(5,l):m_trk(5,m)
1520 m_trk(5,n):m_trk(5,o):m_trk(5,p):m_trk(5,o)
1530 m_trk(5,q)
1540 /*/* M E L O D Y   S y n t h . B R A S S
1550 a=" o4  @v120 18 @79 "/*  M   E   L   O   D Y
1560 b="a1g1d2f2g1
1570 c="c1d1g1&g1
1580 d="c1d1g1c4  d4 r2
1590 e="@v126a1fede2gb<c4>b-2 fg1
1600 f="c2d4f4 g2fega1 g1r
1610 g="agf a2 def a2g4 fede g4.<c1r8
1620 h=">a2gf2 def g4<dg1r r1>
1630 j="ggag4@v126@81>d2efe<d2cdf2>ffec2r8r4
1640 k="<@79@v120ffgf4>@v124 @101a2 g16f16e16d16f16e16d16c16 >a
16g16f16e16dg4.<c1r2r
1650 l="<@v122@79ggag4>@v120@101a2 gfe d4e4f g16a2 g8f16e16d16e
2 r2
1660 m="@v122@79<ffgf4>@v120@101>a2ab<c4 dc4 e4f a2g1 r
1670 n="ffff4raaaa4o4@v117@79r8r8r8
1680 o="@101|:4dfg:|dfg4|:4ega:|ega4
1690 p="|:4a<cd>:|a<cd4|:4dfg:|dfg4>
1700 q="|:4@v115a<cd>:|@v113a<cd4@v110dfgdfgdfg>@79@v124rrrr4r4
rrrf4f2
1710 m_trk(6,a+"r8r1")
1720 m_trk(6,b):m_trk(6,c)
1730 m_trk(6,b):m_trk(6,d)
1740 m_trk(6,e):m_trk(6,f)
1750 m_trk(6,g):m_trk(6,h)
1760 m_trk(6,j):m_trk(6,k)
1770 m_trk(6,l):m_trk(6,m)
1780 m_trk(6,n):m_trk(6,o)
1790 m_trk(6,p):m_trk(6,o)
1800 m_trk(6,q)
1810 /*/*/* B   A   S   S
1820 a=" o3  @v125 18 @70
1830 b="|:16g:||:16b-:|
1840 c="|:16f:||:16g:|
1850 d="|:8g:||:2gg<g>g:||:8b-:||:2b-b-<b->b-:|
1860 e="|:2ffff ff<f>f:| gggg gg<g>g gggg g<g>g<g>
1870 f="ffff4r aaaa4rrr
1880 g="@v127ffff ffff ff<f>f ff<f>f aaaa aaaa aa<a>a aa<a>a
1890 h="@v127gggg gggg gg<g>g gg<g>g<cccc cccc cc<c>c cc<c>c>
1900 q="|:25g:|aaaa4g4 rrr f4f2
1910 m_trk(7,a+"r8r1")
1920 m_trk(7,b):m_trk(7,c)
1930 m_trk(7,b):m_trk(7,c)
1940 m_trk(7,d):m_trk(7,e)
1950 m_trk(7,d):m_trk(7,e)
1960 m_trk(7,d):m_trk(7,e)
1970 m_trk(7,d):m_trk(7,e)
1980 m_trk(7,f):m_trk(7,g)
1990 m_trk(7,h):m_trk(7,g)
2000 m_trk(7,q)
2010 /*/*/* A D P C M  &  F M  H I H A T
2020 a=" o1 @95 @v123 116 p3
2030 b="y2,23r8y2,66r2r y2,23cy2,23c y2,23cy3,3y2,14r4
2040 c="|:3y2,23cc cc y2,14 crrr y2,23cc y2,23cc y2,14 crrr :|
2050 c1="y2,23ccccy2,14crrr ccrry2,14r8y2,14r8
2060 c2="y2,66ccy2,66ccy2,14cry2,14rr y2,14ccy2,65rry2,14r8y2,1
4r8
2070 c3="y2,23ccy2,66ccy2,14cry2,66rr y2,14cy2,14rrry2,14r8y2,6
6r8
2080 d="y2,23cc ccy2,14crrr y2,23cc y2,23cc y2,14crrr
2090 e="y2,14rr y2,14rr y2,14rr y2,14rrrr rr y2,14rr y2,14rr y2
,14rr y2,14rrrry2,23cy2,23c y2,23cr y2,14cc
2100 d1="y2,23rrrry2,14cccc y2,14ccy2,14ccy2,14ccy2,14cc
2110 q="cc cc y2,14rr y2,14rrrry2,14r2
2120 f=c+d
2130 m_trk(8,a):m_trk(8,b)
2140 m_trk(8,f):m_trk(8,c+c1)
2150 m_trk(8,f):m_trk(8,c+c1)
2160 m_trk(8,f):m_trk(8,c+c2)
2170 m_trk(8,f):m_trk(8,c+c3)
2180 m_trk(8,f):m_trk(8,c+d1)
2190 m_trk(8,f):m_trk(8,f)
2200 m_trk(8,e):m_trk(8,f)
2210 m_trk(8,f):m_trk(8,f)
2220 m_trk(8,f+q)
2230 m_play()
```

リスト2 LITTLE WING


```
10 '
20 ' 「 LITTLE WING 」  LINDBERG   By T.ABE
30 '
40 FOR I=0 TO 17
50 READ A$:MEM$(&HB190+I*18,18)=HEXCHR$(A$)
60 NEXT
70 DATA FC 00 4F 4F 14 71 00 00 00 00 1F 1F 1F 1F 00 13 15 11
80 DATA C0 CE 00 11 0F 3F BF 2F 20 30 2C 20 00 C8 80 00 00 32  :
'1 SD
90 DATA C0 00 01 02 00 00 18 32 0A 04 1F 1F 1F 1F 1A 1A 10 90
100 DATA 40 C0 40 00 FD FE F8 F8 2C 20 20 30 00 00 80 00 00 20
:'2 BD
110 DATA FB 00 32 24 52 72 17 13 14 00 1F 1F 1F 1F 09 0A 10 0A
120 DATA 40 C0 83 0A 11 21 35 17 27 20 57 41 00 00 80 00 00 49
:'3 Cymbal
130 DATA F0 00 01 01 01 01 0E 12 17 00 1F 1F 1F 1F 04 1F 1F 1F
140 DATA 00 00 00 00 F6 06 06 08 35 2C 20 20 00 00 80 00 00 2C
:'4 Guitar 1
150 DATA F0 00 01 01 01 01 12 19 17 00 1F 1F 1F 1F 04 1F 1F 1F
160 DATA 00 00 00 00 F6 06 06 08 35 2C 20 20 00 00 80 00 00 2C
:'5 Guitar 2
170 DATA FA 00 01 75 01 41 23 18 2F 03 1E 1F 1C 1F 02 06 03 04
180 DATA 00 01 00 01 15 38 16 06 2C 20 20 30 00 00 80 00 00 20
:'6 Keyboard
190 DATA FB 00 0E 06 07 00 0F 1B 11 04 1F 1F 1F 1F 04 08 16 92
200 DATA 40 40 80 00 32 72 BA F8 20 20 31 2C 00 00 80 00 00 20
:'7 HiHat
210 DATA C8 00 0A 70 30 00 21 29 11 00 1F 1F 5F 5F 12 0E 0A 0A
220 DATA 00 04 04 03 26 26 26 26 31 00 6C 02 00 00 80 00 00 20
:'8 Bass
230 DATA FA 00 61 08 51 02 19 23 22 00 1F 1F 1F 1F 04 0F 1F 1F
240 DATA 01 01 01 01 16 58 03 0A 2C 20 20 30 00 00 80 00 00 20
:'9 Vocal
250 '
260 CSIZE3:CLS:PRINT#0"LITTLE WING / LINDBERG"
270 PRINT"Now MML Setting... Wait a Moment."
280 SCREEN
290 TEMPO0 : PLAY"T170"
300 GOTO 440
310 '--<< PLAY >>--
320 LABEL"!"
330 X$=G$
340 A=INSTR(G$,"I3") : IF A=0 THEN 380
350 PP=(P+1) MOD 2:P$="P"+RIGHT$(STR$(PP+1),1):P=P+1
360 G$=LEFT$(G$,A-1)+"i3@v123"+P$+MID$(G$,A+2,LEN(G$)-A)
370 GOTO 340
380 A=INSTR(G$,"I7") : IF A=0 THEN 410
390 G$=LEFT$(G$,A-1)+"i7p3@v118"+MID$(G$,A+2,LEN(G$)-A)
400 GOTO380
410 PLAY    A$;:PLAY":"+B$;:PLAY":"+C$;:PLAY":"+D$;
420 PLAY":"+E$;:PLAY":"+F$;:PLAY":"+G$;:PLAY":"+H$
430 G$=X$:RETURN
440 '--<< MML DATA >>--
450 A1$="AAAAB&AAB4>C+4.R2<AAAAB&AAA"
460 G1$="CI7CCC CCCC CCCC"
470 H1$="I2CI1CI2C8C8I1C"
480 H2$="I2C8C8I1C"
490 S0$="S2,2,0,12=1H6":S1$="S2,1,0,15=1H5"
500 UP$="s3,1,0,16=3"
510 DN$="s4,1,0,16=3"
520 '
530 A$="I9  V16   Q7 L8 O4 S2,2,0,12=1": 'Vocal
540 B$="I4  V13   Q7 L8 O3"+S0$       : 'Guitar
550 C$="I4  V13   Q7 L8 O3"+S0$       : 'Guitar
560 D$="I6  V12   Q8 L8 O4"           : 'Keyboard
570 E$="I6  V12   Q8 L8 O4"           : 'Keyboard
580 F$="I8  V12   Q7 L8 O4"           : 'Bass
590 G$="I7  @V123 Q6 L8 O6"           : 'Cymbal&Hihat
600 H$="I2  V16   Q6 L4 O1 S4,1,0,12=3": 'SD & BD
610 GOSUB 320
620 B$="ARARBRBD+4EQ1EEQ7EQ1EEEQ7"
630 C$="C+RDRD+RE<G+4AQ1AAQ7AQ1AAA>Q7"
```

```
640 F$="C+RDRD+RE<G+4AAAAAAA>"
650 G$="I706CCCC CCCI3C4I7CCC CCCC
660 H$="I2CI1CI2CI1C8I2CC8I1CI2CI1C
670 GOSUB 320
680 IF DC=1 THEN 710
690 GOSUB 320
700 GOSUB 320
710 H$="I2CI1CI2CI1C8I2CC8I1CC8C8C8C8"
720 GOSUB 320
730 A$="R2F+4AB4A4.F+4F+4F+4.EE4R4"
740 B$=">D<Q1AAAA AAAA AAAA AAAQ7R4ARG+RAR>C+<RARG+AR>D<"
750 C$="AQ1DDDD DDDD DDDD DDDQ7<A>RERERERARERERA"
760 IF DC=1 THEN D$="R1R1R1>A1<":E$="R1R1R1>E1<" ELSE D$="":E$="
"
770 F$="DDDDDDDD DDDDDDDD<AAAAAAAA AAAAAAAA>"
780 G$=STRING$(32,"C")
790 H$="I2CCCCCCCCCCCI1CI2CCCI1C"
800 GOSUB 320
810 B$="R"+RIGHT$(B$,45)
820 C$="R"+RIGHT$(C$,40)
830 F$="DDDDDDDD DDDDDDDD<AAAAAAAA A16.&A+64&B128&>C128&C+EF+AAF
+8&"+DN$+"F+8=0"
840 GOSUB 320
850 A$=A$+"F+2G+4G+4"
860 B$=LEFT$(B$,32)+"A2ARG+R"
870 C$=LEFT$(C$,33)+"E2ERER"
880 D$="":E$=""
890 F$="DDDDDDDD DDDDDDDD <AAAAAAAA AAAAAAG+G+"
900 GOSUB 320
910 A$="G+4.AA2 R2F+4G+4 A2R2 R2AA4A"
920 B$="F+RRF+RRF+RF+RRF+RRF+4>DRRDRRDRDRRDRRD4"
930 C$="C+<Q1F+F+>Q7C+<Q1F+F+>Q7C+<Q1F+>Q7 C+<Q1F+F+>Q7C+<Q1F+F+
>Q7C+4 AQ1DDQ7AQ1DDQ7AQ1DQ7 AQ1DDQ7AQ1DDQ7A4"
940 F$="F+F+F+F+ F+F+F+F+ F+F+F+F+ F+F+F+F+ >DDDDDDDD DDDDDDDD"
950 G$="R4.I3C4.R4 R1I3C4R2.R1"
960 H$="I1CI2C8I1C8I2CCCC8I1A8 I2C I1F4I1CI2CC8I1F8I2CI1F4I2C8I1
C8I2CI1C"
970 GOSUB 320
980 A$="B2.R4R1"
990 B$="E2.RE4EG+&A&G+E<B4"
1000 C$="B2.RB4>K7EG+&A&G+E<B4K5"
1010 F$="EEEE EEEE <EEEE EEEE>"
1020 G$="C4R2.R2.C4"
1030 H$="I1CI2CI1L16AAAAFFFFFFFFDDDDL4I1C8C8C4"
1040 GOSUB 320
1050 A$="G+G+G+G+G+4BA4.R2.A4A4A4>C+4C+4.<BB4R4
1060 B$="Q2G+>EC+<G+>EC+<G+>E< F+>F+C+<F+>F+C+<F+>F+< F+>D<BF+>D
<BF+>D< E>E<BE>E<BE>E<"
1070 C$="R1R1R1R1"
1080 D$="O5G+1A1A1B1"
1090 E$="O5E1F+1F+1G+1"
1100 F$="C+C+C+C+ C+C+C+<F+4 F+EFF+F+G+G+ BBBB BBB>C+<EEFFF+F+G+
G+>"
1110 G$="I3CI7"+STRING$(31,"C")
1120 H$="I2CI1CI2CI1C I2CI1C"+H2$:H$=H$+H$
1130 GOSUB 320
1140 A$="G+4G+4G+4BB4AA4R2AAAF+A4F+A4AR4AA4A"
1150 B$=LEFT$(B$,40)+"Q5>D16D16Q7D2.&DDR4DR4DRQ5E16E16Q7"
1160 C$="R1R2.RQ5A16A16Q7A2.&AAQ1DDQ7AQ1DDQ7ARQ5B16B16Q7"
1170 D$="G+1A1A1&A1"
1180 E$="E1F+1F+1&F+1"
1190 F$="C+C+C+C+ C+C+C+<F+4 F+EFF+F+G+G+>DDDD DDDD DDDD DDDD"
1200 G$=STRING$(32,"C")
1210 GOSUB 320
1220 A$="B1&B2R2"
1230 B$="E4.E2E4Q5EEEE EERQ7"
1240 C$="B4.B2B4Q5BBBB BBRQ7"
1250 D$="B1&B1"
1260 E$="G+2E4F+4G+4AG+&G+2"
1270 F$="EEEE<EEV14"+UP$+"E=0>>A4G+4E8"+DN$+"E4=0<<E4V12"
1280 G$="I3CI7 CCC CCCC CCCCR2"
1290 H$=H1$+H2$+"8I2C8 I1C8C16C16 C16C16C16C16"
1300 GOSUB 320
1310 I=0
1320 ' ｻﾋﾞ
1330 A$=A1$+"4R2."
1340 B1$="DR4.ER4>C+RC+R4C+C+R":B$=B1$+"4<"+B1$+"4<"
1350 C1$="AQ1DDDQ7BQ1DQ7R>ARARQ1DQ7AAR":C$=C1$+"Q1DQ7<":C$=C$+C$
1360 D1$="A4.RDEAC+EAC+EAC+EC+":D$=D1$+D1$
1370 E1$="F+4ERK7DEAC+EAC+EAC+EC+K5":E$=E1$+E1$
1380 F$=">DDDDEEE<A4AAA AAAA>DDDD EEE<F+4F+F+F+F+F+G+&A"
1390 G$=STRING$(32,"C")
1400 H$=H1$+H1$+H1$+H1$+"8I2C8"
1410 GOSUB 320
1420 IF I=1 THEN 1490
1430 ' 1
1440 A$=A1$+"4.R2."
1450 G$="I3CI7"+STRING$(27,"C")+"R2"
1460 H$=H1$+H1$+H1$+H2$+"A8A8C8C8"
1470 GOSUB 320
1480 I=1:GOTO1320
1490 ' 2
1500 IF DC=1 THEN 1610
1510 A$=A1$+"4.R4A4A4B4R4>C+4C+<B4.R2."
1520 B$=B1$+"<Q5D16D16Q7D1&DDR4DDD4E1&EER4EER4"
1530 C$=C1$+"Q5<A16A16Q7A1&AARQ1AQ7AAA4B1&BBRQ1BQ7BBR4"
1540 D$=D1$+"A1&A1B1&B1"
1550 E$=E1$+"F+1&F+1G+1&G+1"
1560 F$=" DDDDEEE<A4AAA AAAA>DDDD DDDD DDDD DDDD EEEE EEEE<EEEE
EEEE>"
1570 G$="I3"+G1$+"CCR4 I3"+G1$+"CCCC CCCC CCCC CCR2."
1580 H$=H1$+H1$+"8I2C8 I2CI1CI2CI1C"+H1$+H1$+H2$+"8I2C8I1C8I2C8I
1C8I2C8"
1590 GOSUB 320
1600 DC=1:GOTO490
1610 ' CODA
1620 A$=A1$+"&"
1630 B$=B1$+"4<DR4.ER4 I5O5V15L8D+&"
1640 C$=C1$+"Q1DQ7<AQ1DDDQ7BQ1DQ7R4"
1650 D$=D1$+"F+2G+4.A&"
1660 E$=E1$+"D2E4.F+&"
1670 F$=" DDDDEEE<A4AAA AAAA>DDDD EEE<F+&"
1680 G$="I3"+G1$+"CCCC CCCC CR4I3C&"
1690 H$=H1$+H1$+"8I2C8 C8C8I1CI2C8I1C8C8I2C8"
1700 GOSUB 320
1710 A$="A4R2.R1"
1720 B$="D+2.&S4,1,0,7=3D+4&S4,1,0,7=3D+2.RO3=0F16&G32&G+32&
1730 C$=""
1740 D$="A4R2. R2I4O4G+G+4A&"
1750 E$="F+4R2.R2I4O4EE4E&"
1760 F$="&F+F+E4F+4R4 F+R4.G+4.A&"
1770 G$=G1$+"CCCI3C&"
1780 H$="R8I2C8I1CI2CI1C"+H1$+"8I2C8"
1790 GOSUB 320
1800 A$=""
1810 B$="A4&=3A2=0B64&>C64&C+8.&C+32&C+4<B8.&A16&B16&>C+16&<B4.
&B8.A16S3,2,0,33=3B4=0>B>F+4E1E&"
1820 D$="AAA4AR4.R2AG+4F+4F+E4F+R4.R2G+G+4A&"
1830 E$="EEE4ER4.R2EE4C+4C+<B4>C+R4.R2EE4E&"
1840 F$="AAG+4AR4.AR4.AG+4F+4F+E4F+R4.F+R4.G+4.A&"
1850 G$=G1$+"R4.I3C&"+G1$+"CCCI3C&"
1860 H$="R8C8I1C"+H2$+H2$+"C8C16C16C8I2CC8I1CI2CI1C"+H1$+"8I2C8"
1870 GOSUB 320
1880 B$="F+16F+16A16F+16 E16F+16A16F+16 A16F+16E16F+16 A16F+16E1
6F+16 A16F+16E16F+16 A16F+16E16F+16 A16&G+16&A&B B32&>C32&C+4.&C
+16<B32&>C32&C+2&C+8.<B32&>C32&C+4.&C+16<B4A16F+32&G64&G+64&"
1890 F$="AAG+4AR4>A4G+4E4&"+DN$+"E4=0<F+4F+E4F+R4.F+R4.G+4.A&"
1900 G$="CI7CCC CCCC RI3C4.C4.C&"+G1$+"CCCI3C&"
1910 H$="R8I2C8I1C"+H2$+"I2C8I1CI2C8I1CI2C8I1C8R8I2C8I1CI2CI1C"+
H1$+"8I2C8"
1920 GOSUB 320
1930 B$="A2&AF+4S3,1,0,7=3E16&S4,1,0,7=3E16=0E2.&EE16&S3,1,0,7=3
E16&=0F+16F+16A16&F+16E16F+16A16&F+16 E16F+16A16&F+16E16F+16A16&
F+16 E8&E32&S3,1,0,7=3E16.&=0E4F+4&S4,1,0,15=3F+4=0"
1940 D$=LEFT$(D$,32)+"R4"
1950 E$=LEFT$(E$,31)+"R4"
1960 F$="AAG+4AR4.AR4.AG+4F+4F+E4F+R4.F+R4.G+2"
1970 G$=G1$+"R4.I3C&"+G1$+"R2"
1980 H$="R8C8I1C"+H2$+H2$+"C8C16C16C8I2CC8I1C"+H1$+"C8C16C16 C16
C16C16C16"
1990 GOSUB 320
2000 B$="D4.C+4.F+4.F4.A4G+4H2S2,1,0,5=1B1&B1=0"
2010 D$=">D4.C+4.D4.C+4.F+4F4G+1&G+1"
2020 E$=""
2030 F$=">D4.C+4.D4.C+4.D4D+4E1&E1"
2040 G$="I3C4.I3C4.I3C4.I3C4.I3C4I3C4I3CI7CCC CCCC CCR2."
2050 H$="I2CI1C8I2CC8I1CI2C8I1C4I2C8I1C8I2C8I1C8I2C8CI1C"+H2$+H2
$+"8I2C8I1C8I2C8I1C8I2C8"
2060 GOSUB 320
2070 A$="AAAAB&AAB4>C+4.R4.<A4AAAB&AAA4.R2."
2080 B$="I4V14Q5O4 DRR2.R2.RD4DDRR2R1"
2090 C$="I4V14Q5O3 ARR2.R2.RA4AARR2R1"
2100 D$="I6V13Q6O5ARR2.R2.RA4AARR2R1"
2110 E$="I6V13Q6O5F+RR2.R2.RF+4F+F+RR2R1"
2120 F$="DRR2.R2.RD4DDRR2R1"
2130 G$="R1R2.RI3C4.R2.R1"
2140 H$="I1CR2.R2.R8I2CC8I1CR2R1"
2150 GOSUB 320
2160 B$="DRR2.R1R2.RQ7<F+4R2RF+R>"
2170 C$="ARR2.R1R2.RC+4<Q1F+F+F+F+F+>Q7C+R"
2180 D$="ARR2.R1R2.RF+&F+1"
2190 E$="F+RR2.R1R2.RC+&C+1"
2200 F$="DRR2.R1R2.RF+4C+C+<F+4F+G+4>"
2210 G$="R1R1R2.RC4I7CCCR2"
2220 H$="CR2.R1R2.R8I2CC8I1CC16C16C8C16C16C8"
2230 GOSUB 320
2240 A$=A1$+"4.R2."
2250 B$="DR4.ER4>C+RC+R4C+C+R4<":B$=B$+B$
2260 C$="AQ1DDDQ7BQ1DQ7R>ARARQ1DQ7AARQ1DQ7<":C$=C$+C$
2270 D$="F+2G+4.A&A1":D$=D$+D$
2280 E$="D2E4.E&E1":E$=E$+E$
2290 F1$="DDDD EEE<A4AAA AAAA>DDDD EE":F$=F1$+"E<F+4F+F+F+F+F+G+
&A>"
2300 G$="I3"+G1$+"CCCC":G$=G$+G$
2310 FOR I=0 TO 2
2320 H$=H1$+H1$+H1$+H1$
2330 GOSUB 320
2340 H$=H1$+H1$+H1$+H2$+"C16C8C16C"
2350 GOSUB 320
2360 NEXT
2370 H$=H1$+H1$+H1$+H1$
2380 GOSUB 320
2390 A$=A1$+"R>C+C+C+EC+4<B&"
2400 B$=LEFT$(B$,42)+"<E&"
2410 C$=LEFT$(C$,62)+"K7<E&"
2420 F$=F1$+"EF+4E4C+4E8&"+DN$+"E4=0"
2430 H$=H1$+H1$+H2$+H2$+"I2C16I1C8.C"+H2$+"8I2C8"
2440 GOSUB 320
2450 A$="BA4F+4.R4R1R1R1"
2460 B$="F+ARF+BR4G+RA2.EF+ARF+B>C+E<F+RBRA2E&"
2470 C$=B$
2480 F$=F1$+"RF+4EC+<B>C+4C+&"+DN$+"C+=0"
2490 HH$=H1$+H2$+H2$:H$=HH$+HH$
2500 GOSUB 320:A$=""
2510 GOSUB 320
2520 GOSUB 320
2530 B$="F+ARF+BR4G+RA2.E&F+ARF+>DC+<BA&A1"
2540 C$="F+ARF+BR4G+RA2.E&F+ARF+>DC+<BE&E1"
2550 F$=F1$+"R<A&A1"
2560 G$="I3"+G1$+"CCCC I3CI7CCC CCCI3C4.R2."
2570 H$=HH$+H1$+"8I2C8R1"
2580 GOSUB 320
2590 END
```

▶湾岸戦争という緊迫した状態の中で，あえて「お笑いマンガ道場」を放送した日本テレビ。
私は日本テレビの車だん吉に対する並々ならぬ思い入れを感じた。　横山　誠(18)群馬県

```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*                                    /*
  30 /*              リ ゾ ・ ラ バ          /*
  40 /*      ( R e s o r t  L o v e r s )  /*
  50 /*                                    /*
  60 /*         B A K U F U - S L U M P    /*
  70 /*                                    /*
  80 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  90 m_sysch("MOP")
 100 m_ycom(1)
 110 /*
 120 m_init()
 130 /*
 140 for i=1 to 19
 150     m_alloc(i,3500)
 160 next
 170 /*
 180 dim char o(40),v(4,10)
 190 dim str  p(16)[256],x,y[50],z[50]
 200 /*
 210 vset()
 220 pdata()
 230 m_play()
 240 end
 250 /*******************************************************/
 260 /*      V O I C E   S E T                              */
 270 /*******************************************************/
 280 func vset()
 290 /*
 300 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
 310 v={  32, 15,  2,  0,195, 40,  0,  0,  0,  3,  0
 320 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 330    , 31,  5,  0,  6, 10, 30,  0,  2,  0,  0,  0
 340    , 31,  5,  0,  6,  4, 35,  0,  0,  6,  0,  0
 350    , 31,  5,  0,  2,  3, 21,  0,  1,  2,  0,  0
 360    , 31,  5,  0,  6,  4,  0,  1,  0,  0,  0,  0}
 370 m_vset(70,v)       /* E.GUITAR
 380 /*
 390             v(0,7)=5:m_vset(72,v)
 400 v(0,4)=205:v(0,7)=4:m_vset(73,v)
 410 /*
 420 /*FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
 430 v={  56, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 440 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMS
 450    , 31,  8,  3,  9, 10, 30,  1,  2,  0,  0,  0
 460    , 31,  3,  3,  9,  3, 25,  1,  0,  0,  0,  0
 470    , 31,  3,  3,  9,  3, 21,  1,  1,  3,  0,  0
 480    , 31, 14, 17,  9, 10,  0,  1,  0,  7,  0,  0}
 490 m_vset(71,v)       /* E.GUITAR (MUTE)
 500 /*
 510 endfunc
 520 /*******************************************************/
 530 /*      S E T   M M L   T O   T R A C K                */
 540 /*******************************************************/
 550 func t(t)
 560     r=0
 570     while o(r)<>255
 580         m_trk(t,p(o(r)))
 590         r=r+1
 600     endwhile
 610 endfunc
 620 /*******************************************************/
 630 /*      P L A Y   D A T A                              */
 640 /*******************************************************/
 650 func pdata()
 660 /*------------------------------------------------------
 670 /*   VOCAL (1.Atmosphere  2.DoctorSolo)  [2][3] * MIDI *
 680 /*     sub (3.DoctorSolo)                [4ch]
 690 /*------------------------------------------------------
 700 x="|:d4c>ar<edc:| d4c>a<cdce& e2
 710 p( 0)="@n2 t144 l8 @38 v12 ro5edc
 720 p( 1)=x+"redc
 730 p( 2)=x+"reee f4e4dc4d4>bb4b<d4c&
 740 p( 3)="c>aa1r2.
 750 p( 4)="|:15r2:|o4e4<d4
 760 p( 5)="|:ddc+c+>bbab4arr|1e4<d4:| >aaa<e&
 770 p( 6)="e>aa4r1aaa<e4.rreeed4c+>bre4<d4
 780 p( 7)="|:e>aa4aaa<e&:| e1 r2c+d4e&
 790 p( 8)="e2reef+ e4dc+4c+ed& d2rdde d4c+>b4b<dc+&
 800 p( 9)="c+2r>aab <c+2r>aa<c+ >b4.bba+b<e& e2redc
 810 p(10)="c>aa2. |:8r1:| r2<c+d4e&
 820 p(11)="|:12r1:|
 830 o={0, 1,2,3, 4, 5,6,5,7, 8,9, 1,2,3
 840           , 4, 5,6,5,7, 8,9, 1,2,10
 850                        , 8,9, 1,1
 860      , 1,2,3, 11, 255}
 870 t(1)
 880 /*
 890 p( 0)="@n3 l8 @45 v11 ro5edc
 900 t(2)
 910 /*
 920 x="|:12r1:|
 930 y="b4afr<c>ba b4agr<c>ba b4aeaba<c& c2
 940 p( 0)="@n4 l8 @45 v10 ro5c>ba
 950 p( 1)=y+"rc>ba
 960 p( 2)=y+"r2 |:4r1:|
 970 p( 3)=x+x
 980 p( 4)="|:15r2:|rc>ba
 990 p( 5)="|:8r1:|
1000 p( 6)=x
1010 o={0,1,2, 3,4,1,2, 3,4,1,2, 5, 4,1,1, 1,2,6,255}
1020 t(3)
1030 /*------------------------------------------------------
1040 /*   KEYBOARD (4-6.Piano & SynBrass)     [5ch]  * MIDI *
1050 /*            ( 7 .Piano)                [6ch]
1060 /*------------------------------------------------------
1070 x="v6r2..o5|:5'a<a>':|'a2<a>' v8
1080 p( 0)="@n5 @2 v8 r2
1090 p( 1)="l1o4 fga2..g8&g aga2..g8&g
1100 p( 2)="fg2..a8 r8a8ar4 @25l8v9b<c+rd&
1110 p( 3)="|:ddc+rdc+r|1d&:|c+4>{br}b2.&
1120 p( 4)="b2b<c+rd&
1130 p( 5)="b2.rr @2
1140 p( 6)=x+x+x+"'e2..<e>''d4<d>''c+<c+>'>'b2.<b>'
1150 p( 7)=x+x+"r1r1r1r2c+dre&
1160 p( 8)="l4|:e.d.e:||:d.c+.d:||:c+.>b.<c+:|>b2..b8& b2b8r.
1170 p( 9)="fg2..a8 r8a8a&l4a.g.
1180 p(10)="<|:3d.c.r.d.cr:| >a1 b2<l8c+dre&
1190 p(11)="b1  <v7c1d1 v9aar2aa r2a4rr
1200 o={0,1,2, 3,4,3,5, 6,7, 8, 1,2
1210          , 3,4,3,5, 6,7, 8, 1,9, 10
1220                          , 8, 1
1230      ,1,2, 3,4,3,11, 255}
1240 t(4)
1250 /*
1260 p( 1)="l1o4|:cde2..e8&e:|
1270 p( 2)="cd2..e8 r8e8er4 @25l8g+arb&
1280 p( 3)="|:bbarbar|1b&:|a4{g+r}g+2.&
1290 p( 4)="g+2g+arb&
1300 p( 5)="g+2.rr @2
1310 p( 6)="l1|:a&aa&a|1a&aa2..g+4g+8g+2.:|aaag+2l8c+drb&
1320 p( 7)="l4b.b.b a+.a+.a+ b.b.b b.b.b
1330 p( 8)="a.a.a a.a.a a2..g+8&g+2g+8r.
1340 p( 9)="cd2..e8 r8e8e&l4e.d.
1350 p(10)="g.g.g.g.gg |:a.a.a.a.aa:| f+1 g+2l8c+drb&
1360 p(11)="g+1  v7a1b1 <v8eer2ee r2eerr
1370 o={0,1,2, 3,4,3,5, 6, 7,8, 1,2
1380          , 3,4,3,5, 6, 7,8, 1,9, 10
1390                        , 7,8, 1
1400      ,1,2, 3,4,3,11, 255}
1410 t(5)
1420 /*
1430 p( 1)="l1o3|:ab<c2..c8&c>:|
1440 p( 2)="ab2..<c8 r8c8cr4 @25l8derf+&
1450 p( 3)="|:f+f+erf+er|1f+&:|e4{dr}d2.&
1460 p( 4)="d2derf+&
1470 p( 5)="d2.rr
1480 p( 6)="l1|:e&ef+&f+|1f+&f+f2..e4e8e2.:|f+fee2l8rrrg&
1490 p( 7)="l4g.g.g |:f+.f+.f+:| f.f.f
1500 p( 8)="e.e.e d+.d+.d+ e2..e8& e2e8r.
1510 p( 9)="ab2..<c8 r8c8c&l4c.>b.
1520 p(10)="<e.e.e.e.ee |:4f+.:|f+f+ f.f.f.f.ff d1 e2l8rrrg&
1530 p(11)="d1  v7f1g1 v9aar2aa r2aarr
1540 t(6)
1550 /*
1560 x="|:dddd dddd eeee|1eee8a8r8d8&:|
1570 p( 0)="@n6 @2 v8 r2
1580 p( 1)="l4o2|:ffffgggg aaaa8<c.ccc>:|
1590 p( 2)="ffffgggl8ga raa1rreard&l4
1600 p( 3)=x+"eeer l1o4
1610 p( 4)="|:c+&c+c+&c+|1d&dc2..>b4b8b2.<:|dc>bb2l4r.<e8&
1620 p( 5)="|:e.d.e:||:d.c+.d:||:c+.>b.<c+:|>b2..e8& e2e8r.
1630 p( 6)="ffff gggl8ga raa1&l4a.b.
1640 p( 7)="o4d.c.>g.<d.c>g |:<d.c.>a.<d.c>a:| e1 e2r.<e8&
1650 p( 8)=x+"eeee
1660 p( 9)="v9f1g1 v8l8aar2aa r2aarr
1670 o={0,1,2, 3, 4, 5, 1,2
1680          , 3, 4, 5, 1,6, 7
1690                   , 5, 1
1700      ,1,2, 8, 9, 255}
1710 t(7)
1720 /*------------------------------------------------------
1730 /*   BASS (8.Fretless)                   [7ch]  * MIDI *
1740 /*------------------------------------------------------
1750 x="ffffffff gggggga raa4&a1
1760 y="dd|:7dr:|>|:6er:|
1770 z="l4q4|:8a:||:8f+:|<dddd
1780 p( 0)="@n7 l8 v12 r2 o2 @72
1790 p( 1)="|:ffffffffgggggggg aaaa<cedc4cccc>egg-:|
1800 p( 2)=x+"ear<d&
1810 p( 3)=y+"ear<d&"+y+"erer
1820 p( 4)=z+"dddd>ffffeeee"+z+">ffffeeee eq8ee8<e>g8&
1830 p( 5)="g2<g8dg8 f+.ec+8>f+8<c+8 >b.<f+>b. <f.c+>bf8
1840 p( 6)="e.<eae8 d+.ad+d+8 e.>el8b<e>e4b<de>errr
1850 p( 7)=x+"&ab4.<
1860 p( 8)="|:16c:||:16d:|>|:16f:| eeee<dde>e eeeee<er>g& l4
1870 p( 9)="l4f.f.f g.ggl8g aar2aa r2aarr
1880 o={0, 1,2, 3, 4, 5,6, 1,2
1890          , 3, 4, 5,6, 1,7, 8
1900                   , 5,6, 1
1910      , 1,2, 3, 9, 255}
1920 t(8)
1930 /*------------------------------------------------------
1940 /*   DRUM sub (9.MelodicTom & DeepSnare) [8ch]  * MIDI *
1950 /*------------------------------------------------------
1960 x="|:7r1:|
1970 y="r1r1r1r1
1980 p( 0)="@n8 l8 r2 o2
1990 p( 1)=x+y+"r2 v15@115car4v13
2000 p( 2)=x+"@114<c>rarfrrr
2010 p( 3)=y+y
2020 p( 4)=x+"@114v10rraar2v13
2030 p( 5)=x+"@115<reereerr>
2040 p( 6)="r1r1r2. v9l16<ccc>v8aaav7fffv6cccv13l8@115arrr
2050 p( 7)=x+y+"@114v8l16r<ccc>aafrr2
2060 o={0,1,2,3,3,4,1, 3,3,5,4,3,6,3, 4,3, 1,7, 255}
2070 t(9)
2080 /*------------------------------------------------------
2090 /*   DRUM (10.BassDrum)                  [10ch] * RHYTHM *
2100 /*        (11.SnareDrum & TomTom)
2110 /*        (12.Closed-HighHat)
2120 /*        (13.CrashCymbal & RideCymbal)
```

▶最近，パチンコで儲けた。そこでひさびさにゲームでも買おうとしたが，引っ越しの資金に回ってしまった。グスン。みんなビンボーが悪いんだ。　澤田　博範(22)神奈川県

```
2130 /*----------------------------------------------------------
2140 x="crrrccrr crrrccrc rccrr1
2150 p( 0)="@n10 18 v15 @u127 r2 o2
2160 p( 1)="|:crrrccrr crrrccrr crrrccrc r2ccrr:|
2170 p( 2)=x+"rrrc
2180 p( 3)="14r|:15c:|
2190 p( 4)="|:62c:|r.c8
2200 p( 5)="18r2rcrc |:6crrrrcrc:| r1
2210 p( 6)="|:61c:|c8c8r.c8
2220 p( 7)=x+"rcrr
2230 p( 8)="|:7crrrccrr:| crrcrrrc
2240 p( 9)="r|:14c:|r  crrr crrr 18ccrrrccc r2ccrr
2250 o={0,1,2, 3,3, 4, 5, 1,2
2260         , 3,3, 6, 5, 1,7,8
2270                 , 5, 1
2280     ,1,2, 3,9, 255}
2290 t(10)
2300 /*
2310 x="|:14rd:|
2320 y="|:6rd:|
2330 p( 0)="@n10 14 v15 @u127 rr o2
2340 p( 1)=x+y+"r1 rr@u115'd8f''d8f'@u127r
2350 p( 2)=y+"rd<c8>d8r8d8
2360 p( 3)="rdrd rdrd rdr
2370 p( 4)="{dr<@u90cc}@u120
2380 p( 5)="{cr>@u90aa}{@u120ar@u90aa}{@u120fr@u90ff}@u127d
2390 p( 6)=x+x+"rdrdrd<c8>d8r
2400 p( 7)=x+"r8<c64>a16..r dr
2410 p( 8)="|:@u127d8@u85{<c>af}8r8:|@u127dd
2420 p( 9)=x+x+"rdrd @u115|:r8'a8<c>''a8<c>':|r@u127
2430 p(10)=x+y+"rrr@133<c64@u110c @u127c64>@u110a
2440 p(11)="@u127a64@u110a @u127f64@u110f @u127d64d8...14r
2450 p(12)=x+"|:r8d64d16..d64d16..:|r
2460 p(13)=x+"rdrd
2470 p(14)=y+"rdr8<c64c16..>a8f8
2480 p(15)="r.<c64c8...c8>a8f8 r<c64c16..>af8d64d8...
2490 p(16)="rd8{<c>af}8rr r16@125<c64c>a64af64fr2
2500 o={0,1, 2,3,4,5, 6, 7, 1
2510       , 2,3,8,   9, 7, 10,11, 12
2520                , 7, 13
2530     ,1, 2,14, 15,16, 255}
2540 t(11)
2550 /*
2560 x="@u95f+@u65f+f+f+
2570 p( 0)="@n10 18 v15        r2 o2
2580 p( 1)="|:20"+x+":|r1r1
2590 p( 2)="|:8r1:|
2600 p( 3)="|:31"+x+":|r2
2610 p( 4)="rr|:53f+:|rr1
2620 p( 5)="|:16"+x+":|
2630 p( 6)="|:15r2:|f+rf+r 14|:rf+f+f+:|r1r1
2640 o={0,1,2,3,4,1, 2,3,4,1,2, 4,5, 1,6, 255}
2650 t(12)
2660 /*
2670 x="rrr.c+8
2680 p( 0)="@n10 14 v15 @u127 rr o3
2690 p( 1)="c+rrrr1"+x+"r1r1r1"+x+"r1
2700 p( 2)="r1 rrr.c+.c+rr"+x
2710 p( 3)="@u65|:7rd+:|r.@u127c+8
2720 p( 4)="@u65|:6rd+:|r2.@u127c+ |:15r1:|"+x
2730 p( 5)="r@u65|:26d+:|d+8@u127c+8 r1
2740 p( 6)="r1 rrr.c+.c+rr rr.c+.
2750 p( 7)="c+8@u40d+8|:27@u90d+8@u40d+8:|@u127"+x
2760 p( 8)="r1r1r1"+x+"r1r1r1r1 |:c+rrr:|18|:3c+c+r2:|
2770 o={0,1,2,3,4,5,1,2, 3,4,5,1,6, 7, 5,1, 1,2,8, 255}
2780 t(13)
2790 /*----------------------------------------------------------
2800 /*   GUITAR (14.16.18.)                    [17-22ch]  * FM *
2810 /*          (15.17.19.) [detune]
2820 /*----------------------------------------------------------
2830 x="o4@71cccc cccc dddd ddd
2840 y="|:ddc+rdc+r|1d&:|c+4>{br}b2.<ar
2850 z="rrdrc+rrd rrc+rdc+rr
2860 p( 0)="@n17 @70 18 r2      y48,20
2870 p( 1)="|:v9"+x+"d eeeeeee@70v11c4eg<@72c&c2v9:|
2880 p( 2)=x+"@70e ree4&e1
2890 p( 3)="b<c+rd&
2900 p( 4)=y+"g+4>b<c+rd&"+y+"ag+&g+2
2910 p( 5)=z+z+z+"rrfrfrre rrereerr
2920 p( 6)=z+z+"rrerdrrd rrfrffrr rr>arg+rra ra&<@73e2.
2930 p( 7)="v11
2940 p( 8)="o3rb<de&e2 e16&f+2... >b<c+dc+>b<f+4. f1
2950 p( 9)=">a4.b4.<c+4 {c+&d+&}32d+32&d+2...
2960 p(10)="e2&eq7f+g+a q6b<c+dd+errr q8
2970 p(11)="&eg4.
2980 p(12)="o5|:5gfe:|g f+>ab<cdef+&a <d1
2990 p(13)=">|:5fed:|f ed+ef+g+ab&<d e1
3000 p(14)="c1d1 eer2ee r2eerr
3010 o={0,1,2,3, 4, 5,6, 7,8,9,10, 1,2,3
3020           , 4, 5,6, 7,8,9,10, 1,2,11, 7,12,13
3030                   , 7,8,9,10, 1
3040     ,1,2,3, 4, 14, 255}
3050 t(14)
3060 /*
3070 p( 0)="@n18 @70 18 r2 r32 y49,00
3080 p( 1)="|:v7"+x+"d eeeeeee@70v9c4eg<@72c&c2v7:|
3090 p( 7)="v9
3100 t(15)
3110 /*
3120 x="o3@71ffff ffff gggg ggg
3130 y="|:@70aaa@71{dd}@70aara&:|ar2..<e@71>{ee}@70<
3140 z="o4rrararra r@71a@70a@71{aa}@70aarr
3150 p( 0)="@n19 @70 18 v9 r2      y50,20
3160 p( 1)="|:"+x+"g aaaa aaar r1:|
3170 p( 2)=x+"@70a raa4&a1
3180 p( 3)="<eera&
3190 p( 4)=y+"e4>rrra&"+y+"ee&e2
3200 p( 5)=z+z+z+"rr<crcrr>b r@71e@70b@71{ee}@70<c+>brr
3210 p( 6)=z+z+"rrararra r@71f@70<c@71>{ff}@70<dcrr r1r1
3220 p( 7)="|:13r2:|r v7 o4q7def+ q6g+aa+bbrrrq8 v9
3230 p( 8)="&ab4.
3240 p( 9)="v10
3250 p(10)="o5|:5edc:|e d>f+gab<cd&f+ a1
3260 p(11)="|:5c>ba<:|c> ba+b<d+ef+g+&a b1
3270 p(12)=">f1g1 aar2aa r2aarr
3280 o={0,1,2,3, 4, 5,6, 7, 1,2,3
3290           , 4, 5,6, 7, 1,2,8, 9,10,11
3300                   , 7, 1
3310     ,1,2,3, 4, 12, 255}
3320 t(16)
3330 /*
3340 p( 0)="@n20 @70 18 v7 r2r64  y51,40
3350 p( 7)="|:13r2:|r v5 o4q7def+ q6g+aa+bbrrrq8 v7
3360 p( 9)="v8
3370 t(17)
3380 /*
3390 x="|:dddrder|1d&:|e4{er}e2.br
3400 y="rrf+rf+rrf+ rrf+rf+f+rr
3410 p( 0)="@n21 @70 18 v9 r2      y52,16
3420 p( 1)="|:11r1:|r2..o4d&
3430 p( 2)=x+"b4eerd&"+x+"bb&b2
3440 p( 3)="rrererre rrereerr"+y
3450 p( 4)=y+"rrararrg+ rrg+rg+g+rr
3460 p( 5)="rrf+rf+rrf+ rraraarr rrererre re&@73b2.
3470 p( 6)="|:8r1:|
3480 p( 7)="|:44r1:|
3490 p( 8)="r1r1r1r1
3500 o={0,1,2, 3,4,3,5, 6, 1
3510       ,2, 3,4,3,5, 7, 1,2, 8, 255}
3520 t(18)
3530 /*
3540 p( 0)="@n22 @70 18 v7 r2 r32 y53,32
3550 t(19)
3560 /*
3570 endfunc
```


リスト4 花

```
 10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
 20 /*                    花                              /*
 30 /*                                                    /*
 40 /* C o m p o s e d   b y   R e n t a r o   T a k i   /*
 50 /*                                                    /*
 60 /*    Programmed by Michikazu Nakanishi(Mii)          /*
 70 /*                                                    /*
 80 /*           i n   D e c e m b e r / 2 5 / 1 9 8 9    /*
 90 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
100 /*
110 m_stop() : m_init()
120 /*
130 for z=1 to 7 : m_alloc(z,2000) : next
140 /*
150 str mp[256],p[256],n[256],mf[256],f[256]
160 /*
170 f="@v125" : mf="@v122" : n="@v119" : mp="@v116" : p="@v113
"
180 /*
190 m_trk(1,"t300 @n20 r64 [d.c.] @b8142 @u122 @ 33 @p 16 q8 o
4 116 @m0 r2..r. [coda] t70")
200 m_trk(2,"t300 @n21 r64 [d.c.] @b8242 @u122 @ 33 @p112 q8 o
4 116 @m0 r2..r. [coda] t70")
210 m_trk(3,"t300 @n22 r64 [d.c.] @b8192 @u108 @  3 @p 64 q8 o
4 116 @m0 r2..r. [coda] t70")
220 m_trk(4,"t300 @n22 r64 [d.c.] @b8192 @u108 @  3 @p 64 q8 o
4 116 @m0 r2..r. [coda] t70")
230 m_trk(5,"t300 @n22 r64 [d.c.] @b8192 @u108 @  3 @p 64 q8 o
4 116 @m0 r2..r. [coda] t70")
240 m_trk(6,"t300 @n23 r64 [d.c.] @b8092 @u108 @  3 @p 64 q8 o
3 116 @m0 r2..r. [coda] t70")
250 m_trk(7,"t300 @n24 r64 [d.c.] @b8292 @u108 @  3 @p 64 q8 o
2 116 @m0 r2..r. [coda] t70")
260 /*
270 m_trk(1,"r2 r2 r2 r2")
280 m_trk(2,"r2 r2 r2 r2")
290 m_trk(3,"@v113g>b<@v114g>b<@v115ad@v116bd @v117<c>e@v118<e
>e<d>e@v117<c>e @v116bd@v115g>b<@v114ac@v113ac q4g8g8g8r8q8")
300 m_trk(4,"@v113dr@v114dr@v115f+r@v116gr @v117ar@v118grar@v1
17ar @v116gr@v115dr@v114f+r@v113f+r q4d8d8d8r8q8")
310 m_trk(5,"r2 r2 r4@v114dr@v113dr >q4b8b8b8r8<q8")
320 m_trk(6,p+"g4.f+8 e8c8>a4 <d4d4 q4g8g8g8r8q8")
330 m_trk(7,p+"g4.f+8 e8c8>a4 <d4d4 q4g8g8g8r8q8")
340 /*
350 m_trk(1,mf+"d8.dg8rg agf+ed4 >b<dgab8<d8 >a4.r8")
360 m_trk(2,mf+"d8.d>b8rb <cedc>b4 b8b<dg8g8 f+4.r8")
370 m_trk(3,mf+"|:4g>b<:||:4g>g<:||:3g>b<:|g>g <f+>a<f+>a<e>a<
f+>a<")
380 m_trk(4,mf+"drdrdrdr ererdrdr drdrdrdr drdrdrdr")
390 m_trk(5,mf+"r2 r2 r2 r4crcr")
400 m_trk(6,mf+"g2 c4>b4 <g2 r2")
410 m_trk(7,mf+"g2 r2'g4.b8 <d4>d4")
420 /*
430 m_trk(1,mp+"d8.dg8rg agef+d4 a8.aa8b8 g4.r8")
440 m_trk(2,mp+"d8.d>b8rb <cedc>b4 a8.<f+f+edc >b4.r8<")
450 m_trk(3,mp+"|:4g>b<:||:4g>g<:||:4f+>a<:|gd>b<d>@v118g<q4@v
120d@v122g@v124bq8")
460 m_trk(4,mp+"|:4dr:| ererdrdr |:4dr:| >brrrr4")
```


▶ある日，電車の車内放送で「火薬，危険物の持ち込みは禁止されております」。テロリストは全員日本語がわかるのだろうか。　村上　淳一(19)福岡県

```
 470 m_trk(5,mp+"|:4r2:|")
 480 m_trk(6,mp+"g2 c4>b4< d4&dc+d>d g4.r8<")
 490 m_trk(7,mp+"g2 r2 r2 r2")
 500 /*
 510 m_trk(1,f+"<d8.dd8r>b <c8.cc8r>a b4e8a8 d4.r8")
 520 m_trk(2,f+"g8.ab8rg a8.gf+8rf+ d4d8c+8 d4.r8")
 530 m_trk(3,f+"<d>d<d>d<d>dbd <c>d<c>d<c>dad bdbca>b<g>a <f+>a
<f+>a<f+8r8")
 540 m_trk(4,f+"<brbrbrgr arararf+r grgrerer drdrd8r8")
 550 m_trk(5,f+"grgrgrrr f+rf+rf+rrr r2 r2")
 560 m_trk(6,f+"g2 d2 g8e8c8c+8 d4&dq4@v124c>@v123b@v122a<q8")
 570 m_trk(7,f+"g2 d2 g8e8c8c+8 d4r4")
 580 /*
 590 m_trk(1,mf+"d8.dg8g8 agf+ed4 a8a.b32<c8>a8 g4.r8")
 600 m_trk(2,mf+"d8.d>b8b8 <cedc>b4 a8<d8d8ef+ g4.r8")
 610 m_trk(3,mf+"|:4g>b<:||:4g>g<:||:4f+>a<:|gd>b<d>grrr")
 620 m_trk(4,mf+"drdrdrdr ererdrdr drdrdrdr >b<rrrr4")
 630 m_trk(5,mf+"|:4r2:|")
 640 m_trk(6,"g2 c4>b4< d4>d4 g4&g132<dgb<dgbl16>")
 650 m_trk(7,"g2 |:3r2:|")
 660 /*
 670 m_trk(1,"|:4r2:|")
 680 m_trk(2,"|:4r2:|")
 690 m_trk(3,f+"<<d8..d32d8>b8 <c8..c32c8>a8 b4q4e8a8q8 d4@v121
e8@v119f+8")
 700 m_trk(4,f+"b8..b32r4 a8..a32r4 g4>q4a8<e8 q8>a4<@v121c4")
 710 m_trk(5,f+"gdgdbdgd f+df+dadf+d d4r8q4>a8 q8f+4@v121f+4<")
 720 m_trk(6,f+"g2 d2 q4g8e8c8c+8 q8d4@v121d4")
 730 m_trk(7,f+"g2 d2 q4g8e8c8c+8 q8d4@v121d4")
 740 /*
 750 m_trk(1,p+"|:4r2:|")
 760 m_trk(2,p+"d8.dg8rg agf+ed4 >b<dgab8g8 a4.r8")
 770 m_trk(3,p+"g>g<|:3g>b<:| |:4g>g<:| |:3g>b<:|g>g< f+>a<f+>a
<e>a<f+>a<")
 780 m_trk(4,p+">br<drdrdr ererdrdr drdrdrdr drdrdrdr")
 790 m_trk(5,p+"|:3r2:| rrrrcrcr")
 800 m_trk(6,p+">g2 <c4>b4 <g2 r2")
 810 m_trk(7,p+">g2 r2 <g4.b8 <d4>d4")
 820 /*
 830 m_trk(1,n+"|:4r2:|")
 840 m_trk(2,n+"d8.dg8rg agef+d4 a8.aa8b8 g4.r8")
 850 m_trk(3,n+"|:4g>b<:||:4g>g<:||:4f+>a<:|gd>b<d>g<q4@v120d@v
121g@v122bq8")
 860 m_trk(4,n+"drdrdrdr ererdrdr drdrdrdr >b<rrrr4")
 870 m_trk(5,n+"|:4r2:|")
 880 m_trk(6,n+"g2 c4>b4< d4&dc+d>d g4.r8<")
 890 m_trk(7,n+"g2 |:3r2:|")
 900 /*
 910 m_trk(1,f+"<d8.dd8r>b <c8.cc8r>a b4@v124e8@v123a8 @v122d4.
r8")
 920 m_trk(2,f+"|:4r2:|")
 930 m_trk(3,f+"|:3<d>d:|bd |:3<c>d:|ad bdbca>b<g>a< f+>a<f+>a<
f+8r8")
 940 m_trk(4,f+"brbrbrgr arararf+r grgrerer drdrd8r8")
 950 m_trk(5,f+"grgrgrrr f+rf+rf+rrr r2 r2")
 960 m_trk(6,f+"g2 d2 g8e8c8c+8 d4&d@v124q4c>@v123b@v122aq8<")
 970 m_trk(7,f+"g2 d2 g8e8c8c+8 d4r4")
 980 /*
 990 m_trk(1,mf+"d8.dg8g8 agf+ed4 a8a.b32<c8>a8 g4.r8")
1000 m_trk(2,mf+"|:4r2:|")
1010 m_trk(3,mf+"|:4g>b<:||:4g>g<:||:4f+>a<:|gd>b<d>grrr<")
1020 m_trk(4,mf+"drdrdrdr ererdrdr drdrdrdr >b<rrrr4")
1030 m_trk(5,mf+"|:4r2:|")
1040 m_trk(6,mf+"g2 c4>b4 <d4>d4 g4&g132<dgb<@v123d@v124g@v125b
116>")
1050 m_trk(7,mf+"g2 |:3r2:|")
1060 /*
1070 m_trk(1,"|:4r2:|")
1080 m_trk(2,"|:4r2:|")
1090 m_trk(3,f+"<d8..d32dcc>b< c8..c32c>bba b4q4e8a8 q8d4r4")
1100 m_trk(4,f+"b8..b32baag a8..a32aggf+ g4>q4a8<e8 q8>a4r4<")
1110 m_trk(5,f+"gdgdd4 f+df+dd4 d4r8q4>a8 q8f+4r4<")
1120 m_trk(6,f+"g2 d2 q4gf+edc8c+8 q8d4&dc@v126>a<@v127d")
1130 m_trk(7,f+"g2 d2 q4gf+edc8c+8 q8d4&dc@v126>a<@v127d")
1140 /*
1150 m_trk(1,f+"d8.dg8rg agf+ed4 >b<dgab8<d8 >a4.r8")
1160 m_trk(2,f+"d8.d>b8rb< cedc>b4 b8.<dg8g8 f+4.r8")
1170 m_trk(3,f+"|:4g>b<:||:4g>g<:||:3g>b<:|g>g< f+>a<f+>a<e>a<f
+>a<")
1180 m_trk(4,f+"drdrdrdr ererdrdr drdrdrdr drdrdrdr")
1190 m_trk(5,f+"|:3r2:| rrrrcrcr")
1200 m_trk(6,f+">g2 <c4>b4< g2 r2")
1210 m_trk(7,f+">g2 r2 <g4.b8< d4>d4")
1220 /*
1230 m_trk(1,mf+"d8.dg8g8 agef+d4"+p+"a8.aa8b8 g4.r8")
1240 m_trk(2,mf+"d8.d>b8b8< cedc>b4"+p+"a8.<f+f+edc >b4.r8<")
1250 m_trk(3,mf+"|:4g>b<:||:4g>g<:|"+p+"|:4f+>a<:|gd>b<d>gq4"+m
p+"<d"+n+"g"+mf+"b")
1260 m_trk(4,mf+"drdrdrdr ererdrdr"+p+"drdrdrdr >brrrr4")
1270 m_trk(5,"|:4r2:|")
1280 m_trk(6,mf+"g2 c4>b4<"+p+"d4&dc+d>d g4.r8<")
1290 m_trk(7,mf+"g2 r2 r2 r2")
1300 /*
1310 m_trk(1,f+"<d8d8d8.>b <c8c8c8r>a b4e8a8 d4.r8")
1320 m_trk(2,f+"g8a8b8.g a8g8f+8rf+ d4d8c+8 d4.r8")
1330 m_trk(3,f+"|:3<d>d:|bd |:3<c>d:|ad bdbca>b<g>a <f+>a<f+>a<
f+8r8")
1340 m_trk(4,f+"<brbrbrgr arararf+r grgrerer drdrd8r8")
1350 m_trk(5,f+"grgrgrrr f+rf+rf+rrr r2 r2")
1360 m_trk(6,f+"g2 d2 g8e8c8c+8 d4&dq4@v124c@v123>b@v122aq8<")
1370 m_trk(7,f+"g2 d2 g8e8c8c+8 d4r4")
1380 /*
1390 m_trk(1,mf+"d8.d@v123g8@v124g+8 @v125ab<@v126cd@v127e2r8 d
8.d>a8b8 g4.r8")
1400 m_trk(2,mf+"d8.c@v123>b8@v124b8 <@v125ce@v126ab<@v127c2r8
@v125>b8agf+edc >b4.r8")
1410 m_trk(3,mf+"d>g<f+>a<@v123g>b<@v124g+>b @v125<a>a@v126<a>a
<@v127g2@v125g8 g>b<g>b<f+>a<f+>a<"+mf+"g>b@v123<g>b<@v124ad@v12
5bd")
1420 m_trk(4,mf+">br<dr@v123er@v124er @v125er@v126er@v127e2@v12
5e8 drdrdrdr"+mf+"dr@v123dr@v124f+r@v125gr")
1430 m_trk(5,mf+"rrcrr4 r4@v127>g2@v125b-8 r2"+mf+"r2")
1440 m_trk(6,mf+"l8gf+@v123e@v124d @v125c4@v127c2@v125c+ d4
   >d4        "+mf+"   g4.@v125f+")
1450 m_trk(7,mf+"l8gf+@v123e@v124d @v125c4@v127c2@v125c+ t65d&t
60d>t55d&t50d"+mf+"t70g4.@v125f+")
1460 /*
1470 m_trk(1,"|:3r2:| |:2r1:|[*]")
1480 m_trk(2,"|:3r2:| |:2r1:|[*]")
1490 m_trk(3,"@v126<c>e@v127<e>e<d>e@v126<c>e @v125bd@v124g>b<@
v123ac@v122ac @v121g4.r8 |:2r1:|[*]")
1500 m_trk(4,"@v126ar@v127grar@v126ar @v125gr@v124dr@v123f+r@v1
22f+r @v121d4.r8 |:2r1:|[*]")
1510 m_trk(5,"r2 r4@v123dr@v122dr @v121>b4.r8 |:2r1:|[*]")
1520 m_trk(6,"@v126e<@v127c>a4 @v125<d4>@v123d4        @v121q4
 g@v120   d>@v119   gr |:2r1:|[*]")
1530 m_trk(7,"@v126e<@v127c>a4 @v125<d4>@v123t65d&t60d @v121q4t
55g     t50r       t45rr |:2r1:|[*]")
1540 /*
1550 /*  やっとおわった。つかれたぁぁ。  さあ、えんそうよーーん
。
1560 /*
1570 m_play()
1580 /*
1590 /*  それではみなさん、さよおぬわるわーーーーー！！！
1600 /*
```

ZMUSIC.FNC変更のお知らせ

どうも，このコーナーは久し振りの（善）です。ええ，大変恐縮なんですが，'90年10月号掲載の「ZMUSIC.FNC」にバグが見つかったのでそのアフターケアをします。ソースリストで入力された方は「方法A」で，ダンプで入力された方は「方法B」に従ってデバッグを行ってください。なお，「方法B」の方は'90年6月号付録ディスクに収録されている「dis.x」とアセンブラ，リンカが必要です。

方法A

まず10月号のP.107の左側の349行と350行の命令を削除してください。

次に10月号のP.109の右側の1039行と1040行の間に以下の命令群を挿入してください。

```
cmpi.b    #$43,d0
bne.s     exit_wr_kc
tst.w     d7
beq.s     exit_wr_kc
move.b    #$3e,(a2)+
```

以上です。あとは10月号の手順どおりアセンブル・リンクを行ってください。

方法B

dis.xが準備できたらこれを使って「zmusic.fnc」のソースリストを以下のようにして作成してください。

A>dis zmusic.fnc zmusic.s

無事ソースリストが作成されたらそれをエディタで読み込んで，289行付近から以下の命令群を見つけ出してください。なおラベル（「L000d67」など）は下に示したものと違う場合もあります。見付けたら「網掛け」の部分の命令を削除してください。

```
tst.b     L000d67
bmi       L000394
sub.w     d5,L000e02
bmi       L000394
move.b    #$40,(a2)+
move.b    #$4c,(a2)+
```

次に同じリスト中の897行付近から以下に示したような命令群を見つけ出してください。

```
cmpi.b    #$20,d0
beq       L000a0a
move.b    d0,(a2)+
bra       L000a0a
```

「L000a0a」のラベルはあなたが「dis.x」で作成したソースリストとは違う場合もあります。以下「L000a0a」相当するラベルを「ラベル」と表記することにします。さて，うまく見つけ出せたら「move.b　d0,(a2)+」と「bra　L000a0」の間に以下の命令群を挿入してください。

```
cmpi.b    #$43,d0
bne       ラベル
tst.w     d7
beq       ラベル
move.b    #$3e,(a2)+
```

挿入が完了したら10月号の手順に従ってアセンブル・リンクを行ってください。

以上です。

▶「C-TRACE」のためにゲームを買えないこの俺に，ほぼ1年ぶりに買いたいゲームが登場した。「栄冠は君に」，うーん，ヤバイ。C-TRACE/TPが30分の1ほど遠のいていく。ゲームレビューなんて読んだのがまずかった。　下田　達也(23)三重県

ようこそここへC言語

配列って何だろう(その2)

[第5回]

Nakamori Akira
中森 章

前回は一次元配列を中心に，基本的な概念やその使い方を紹介しました。一次元の配列がわかれば二次元以上でも基本は同じです。今回は二次元の配列を使って生物モデルのシミュレーションとしてお馴染みのライフゲームを作ってみましょう。

「パロディウスだ！」のX68000版の発売日を心待ちにしながら，毎日ゲームセンターで練習に励んでいる中森章です。私はシューティングゲームは苦手なのですが「パロディウスだ！」だけはキャラが可愛くて妙に愛着を感じてしまいます（ファミコン版も買ってしまった)。

さて，今回も前回に引き続いて配列を取り上げます。前回はプログラミングの基本ともいえる一次元配列について説明しましたが，今回は二次元以上の配列について説明します。

二次元の配列

二次元の配列とは2つの添字で要素を参照する配列のことです。ここで，前回の一次元配列の例題を思い出してみましょう。前回は何人かの受験生のある試験の得点が格納されている配列を考えてみました。配列の名前をmarkとすると，

mark[0]は受験番号1の人の得点
mark[1]は受験番号2の人の得点
mark[2]は受験番号3の人の得点
⋮
mark[N]は受験番号（N+1）の人の得点

という具合に得点と（受験番号で代表される）人の関係を配列で表すことができます。これは配列要素を受験番号という添字で参照していることになります。さて，この例ではある特定の科目の試験の得点しか表現することができません。しかし，ある試験の得点を扱う場合，特定の1科目だけではなく複数の科目を同時に扱いたいこともあります。

まず，一次元配列しか使えない場合にどうなるか考えましょう。複数の科目，たとえば，英語，数学，国語の試験の得点を一次元配列で表すと，各科目に対応して3つの一次元配列が必要です。それを，eigo，sansuu，kokugoという名前の配列であるとしておきます。もし，これで受験番号（N-1）の人についてそれぞれの科目の試験の得点の合計goukeiを求める場合は，

goukei=eigo[N]+sansuu[N]+kokugo[N]；

という式で計算できますね。ところが，試験の科目数が100や1000もあったらどうでしょう（現実にははありえない？)。100種類以上の配列名を考えるのも面倒ですが，試験の合計を求める式を書くのも面倒です[1)]。

そこで登場するのが二次元配列なのです。一次元配列では“受験番号”というただひとつの添字で要素を参照していました。ところが，二次元配列ではもうひとつ添字を用いて合計2つの添字で要素を参照することができます。この例では“科目”と“受験番号”という2つの添字を使えばよいのです。もちろん，添字として使用できるのは整数値だけですから，

添字0 ⟷ 英語
添字1 ⟷ 数学
添字2 ⟷ 国語
⋮

などという対応はあらかじめ決めておかなければなりません。このような二次元配列を使用すると，先に示した3教科の合計点は，繰り返し制御構造を用いて，

```
goukei=0；
for(i=0；i<3；i=i+1)
  goukei=goukei
    + [科目i，受験番号Nで参照できる要素]
```

と簡潔に表すことができます。これは科目数がどのように増えても同じように表現できるでしょう。

今は，二次元配列を2つの添字で参照できる配列として考えました。これは数学でいう行列のイメージです。つまり平面上に行を表す項目と列を表す項目を直交させて配置し，行の項目と列の項目の格子点で要素を示すというものです（図1)。ここで二次元配列をもう少し別の局面から考えてみましょう。“科目”と“受験番号”で参照できる配列があるということは，“受験番号”で参照できる一次元配列（これはある1科目の得点が格納されている）が“科目”の数だけあるのと同じことになります。逆に見ると，“科目”を指定して参照する配列

図1 二次元配列の図形的イメージ（その1）

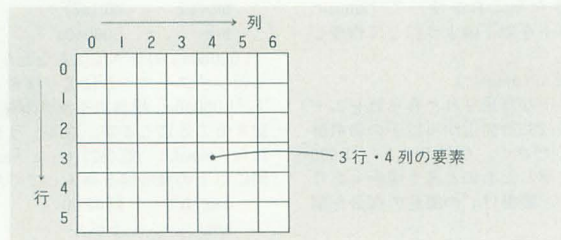
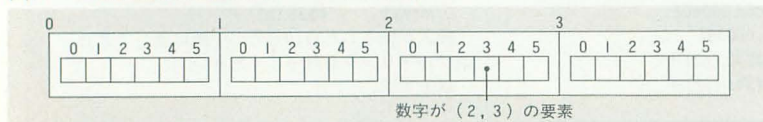



図2 二次元配列の図形的イメージ（その2）

の各要素は“受験番号”を指定して参照する一次元配列と見なすこともできます。つまり，“科目”と“受験番号”で参照する二次元配列とは“科目”を指定して参照する一次元配列であり，その要素がさらに“受験番号”を指定して参照する一次元配列ということになります(図2)。

二次元配列に関しては行列として理解しても一次元配列を要素とする一次元配列として理解してもプログラムを書く上では何の問題もありません。しかし，C言語での配列の表記の受け入れやすさや次元を三次元以上に拡張する場合の理解のしやすさを考えると，後者の考え方のほうがいいと思われます[2]。

---

1) 一次元配列の説明のときも同じようなことをいった記憶がある。要するに，配列とは複数の変数や配列をひとまとめにして扱うためのデータ構造なのである。

2) とはいえ，2種類の理解をしておき，時と場合に応じて考え方を変えるのがベスト。

C言語で二次元配列を扱う

それでは，C言語における二次元配列の使用法を具体的に説明しましょう。一次元配列のときも説明しましたが，C言語で配列を扱うためには，

- **配列の宣言**
- **配列要素の参照**
- **配列の用途**

を押さえなければなりません。以下に順次説明していきましょう。

●配列の宣言

二次元配列の宣言は，一次元配列と同様に，配列要素のデータ型と添字の大きさを指定します。二次元配列では2つの添字がありますから，

データ型　名前［定数式］［定数式］；

という形式で，添字の大きさを示す［と］の組を2つ指定して宣言します。左側の［ ］内の定数式が二次元配列の要素である一次元配列が何組あるかを指定し，右側の［ ］内の定数式が二次元配列の要素[3]である一次元配列の大きさ（要素数）を指定します。たとえば，

```
float fpa[20][30];
```

という表現は要素数30の一次元配列が20個集まっているfloat型の配列fpaを宣言していることになります。また，一次元配列の場合とまったく同様に，

名前［定数式］［定数式］

の部分をカンマ（，）で区切って並べることで，同じデータ型の要素を持つ複数の二次元配列を一度に宣言することも可能です。また，カンマで区切ることで，同時に同じデータ型の一次元配列や変数の宣言をすることもできます。たとえば，

```
float fpa[2][3], fpb, fpc[2], fpd[3][3];
```

という具合です。

●配列要素の参照

二次元配列は2つの添字を指定して参照します。ただし，二次元配列の要素を参照するための特別な演算子というものはなく，一次元配列と同じ演算子を使用します。具体的には，配列要素を参照するための演算子［ ］を配列名のあとに2個付けることで2つの添字を指定します。たとえば，

```
float fpb[2][3];
```

と宣言されている二次元配列は，

```
fpb[0][0] fpb[0][1] fpb[0][2]
fpb[1][0] fpb[1][1] fpb[1][2]
```

という表現で参照できる6個の要素を持っていることになります。もし，fpb[0][1]の値を1にしたいのであれば，

```
fpb[0][1]=1;
```

という式を書けばいいですし，逆にこの要素の値を別の変数xに代入したいのであれば，

```
x=fpb[0][1];
```

という式を書けばいいでしょう。これは変数や一次元配列の参照法と同じです。

ところで，上の6個の要素を注意深く見てみましょう。それらは，

```
(fpb[0])[0]   (fpb [0])[1]   (fpb[0])[2]
(fpb[1])[0]   (fpb [1])[1]   (fpb[1])[2]
```

と読み変えることができますね。今，

```
fpb[0]  →  A
fpb[1]  →  B
```

と置き換えることにすると，

```
A[0]    A[1]    A[2]
B[0]    B[1]    B[2]
```

となって，二次元配列fpbは要素数が3である2つの一次元配列から成り立っていることがわかります。また置き換え前の，

```
fpb[0]  fpb[1]
```

というのも明らかに一次元配列を意味しています。これは一次元配列を要素とする一次元配列である二次元配列の性格をよく表しているものです。話が飛びすぎたので，少し詳しく説明します。そもそもC言語には二次元配列という概念はあるものの，配列要素を参照する手段（演算子）としては一次元配列としての参照しかできないようになっているのです。そこで二次元配列を一次元配列の一次元配列と読み変えて2段階の演算（［ ］を2回使う）で配列要素を参照するのです。FORTRANやPASCALやBASICでの二次元配列の要素の参照を思い出してみましょう。

A (3, 4)　　　A[2, 3]

というように2つの添字をカンマ（，）で区切って一度の操作で配列要素を参照します。ここには二次元配列を一次元配列の一次元配列と見なすという考えなど存在していません。

●二次元配列の用途

さて，二次元配列の用途について考えてみましょう。二次元配列といっても基本は一次元配列（の集まり）ですから，一次元配列が用いられる用途（統計処理，数表）にはもちろん適用できます。また，二次元という空間的なイメージから，視覚的に格子状をしたデータを表すためにも使用されます。たとえば，将棋，チェス，オ

セロなどの盤面のデータ，あるいはもっと一般的なX座標とY座標を持った平面のデータを表すこともできるでしょう。

3) すでに，二次元配列を，その要素が一次元配列である一次元配列と見なして話を進めている。

二次元配列の初期化

配列を表として使用する場合など，配列要素にあらかじめ初期値を与えておきたいということがあります。ここでは二次元配列の要素を初期化する方法について説明します。

二次元配列の初期化は一次元配列の場合と同様です。すなわち，配列を宣言するとき，宣言のあとに配列要素をカンマ(，)で区切った初期値のリストを｛と｝で囲み，＝でつなぎます。このとき重要なことは配列要素の順序です。一次元配列の場合は初期値を添字の順に並べるだけでしたが，二次元配列には2つの添字があります。この添字の順序関係を知っていなければ期待したような初期化を行うことはできません。

ここで思い出してほしいのは，二次元配列とは要素を一次元配列とする一次元配列であるということです。こう考えると二次元配列の初期化も一次元配列の初期化と同じような方法で行うことができます。いま，

```
int ia[3][4];
```

と宣言される二次元配列を初期化することを考えましょう。これは，まず，

ia[0]　ia[1]　ia[2]

という3つの一次元配列を要素とする一次元配列です。これが，それぞれ要素数4の一次元配列になっているのでしたね。そのため初期化は，

int ia[3][4]＝{ □，□，□ }；

という形式をしていなければなりません。これは要素が3つの一次元配列を初期化するやり方です。すなわち，左の□がia[0]の初期値，真中の□がia[1]の初期値，右の□がia[2]の初期値で，それぞれ要素数が4の一次元配列の初期値となっているのです。したがって，それぞれの□の部分を，さらに，

{●，●，●，●}

という形で初期化すればよい（一次元配列の初期化）ことがわかります。

このとき配列iaを初期化する｛ ｝の中に，その要素である一次元配列の初期化リストを｛と｝で囲んだ3組の初期値がカンマ(，)で区切って入ることになります。したがって，iaという二次元配列は，たとえば，

```
int ia [3][4]= {
    {1,2,3,4} ,
    {10,20,30,40} ,
    {1,3,5,7}
};
```

というように｛と｝の組を入れ子にして初期化できるのです。どの初期値がどの配列要素に対応するかはわかりますね。例を挙げるとia[1][3]は一次元配列ia[1]の添字3（4番目）の要素ですから，{10,20,30,40}の4番目の要素である40ということになります。このように，二次元配列でも一次元配列を要素とする一次元配列と考えることで初期化の形式をすんなりと理解することができるのです。

ところが，C言語では二次元配列を初期化する方法（形式）はこれだけではありません。さすが，なんでもあり（というか，いいかげんな）のC言語です。二次元配列は一次元配列を要素とする一次元配列と見なすこともできますが，先に説明したように，単純な行列と見なすこともできるのです。このとき，要素を参照するときの左側の［ ］が行を示し，右側の［ ］が列を示します。

たとえば，

ia[2][3]

という表現はiaという行列の第2行，第3列の要素（行，列とも第0から始まるとき）を示していることになります。このような考え方をする場合，要素が一次元配列云々といってもしようがありません。この場合には二次元配列の要素は行ごとに列の順序で格納されるということを覚えておかなければなりません（行の小さい順，列の小さい順に格納される)。これはFORTRANやBASICでの二次元配列の初期化[4]と同じく，ただ暗記だけにたよる方法であり，あまり発展性がないと思うのですがどうでしょう。とにかく，二次元配列を行列と見なした場合の初期化は，

```
int ia[3][4]= {
    1, 2, 3, 4,
    10,20,30,40,
    1, 3, 5, 7
};
```

となり，外側の｛ ｝の中に｛や｝は使いません。これは一次元配列の一次元配列と考えた場合の記述で｛と｝が省略された記法と見ることもできますね[5]。初期値の省略がない場合，どちらでも初期値の並ぶ順序は同じになります。

4) FORTRANでは二次元配列の要素は列の順序で格納される。他の言語は行の順序で格納されることが多い。統一されてないのでうろ覚えだと混乱を招く。

5) 厳密には｛と｝を省略したものが同じ意味になるとは限らない。二次元配列の要素を一次元配列の一次元配列と見なした場合は，要素である一次元配列の初期値に省略が使える。たとえば，

int ia[2][3]＝{ {1},{3} }；

という表現はia[0][0]を1にia[1][0]を3に初期化するが，

int ia[2][3]＝{2,3}；

はia[0][0]を1にia[0][1]を3に初期化する。

二次元より大きな次元の配列

C言語では次元が2以上の三次元配列，四次元配列，…，といった配列を使用することができます。実際のプログラムではこのような次元の大きい配列が必要になることはあまりありませんが，三次元配列は二次元配列を要素とする一次元配列，四次元配列は三次元配列を要素とする一次元配列，…，n次元配列は（n－1）次元配列を要素とする一次元配列と考えれば，なにも恐れるこ

とはありません。配列の宣言や要素を参照するときに［　］という演算子が次元の数だけ現れてくるだけで，本質はすべて一次元配列なのですから。

◆基礎力を高めよう

設問1　次に示す二次元配列の大域宣言があるとき1）から9）に示す配列要素の値を答えてください。

```
配列宣言    int x[3][4]={
                {1,2},
                3,4,5,6,7,8,9
            };
```

1)　x[0][0]
2)　x[0][1]
3)　x[0][2]
4)　x[1][2]
5)　x[1][3]
6)　x[2][0]
7)　x[2][3]
8)　x[2][−1]
9)　x[0][4]

（解答は60ページ）

二次元配列を扱うプログラム

それでは，二次元配列を使った実際のプログラムを紹介しましょう。ここでは典型的な配列の用途に従って3つのプログラムを作ってみます。すなわち，前回の一次元配列の例題を拡張した統計処理（もどき）のプログラム，平面上のデータを扱うライフゲームのプログラム，および二次元配列を（数学の）二次行列と同一視して連立方程式を解くプログラムです。以下に簡単な解説をしておきますが，どのプログラムもそれほど難しいことはやっていませんから，各自解読してみてください。

●統計処理もどき

与えられた配列に何人かの試験の得点が科目別格納されている場合に，各科目の得点の平均と標準偏差を計算し，さらに各自の得点の偏差値を計算するプログラムがリスト1です。これは前回に示した例題の拡張版です。得点はあらかじめmarkという名前のint型二次元配列の中に初期値として格納されているとします。このとき，

mark[n][m]

という要素は受験番号mの人の科目nの得点としています。また，偏差値は前回と同じく，

(mark[n][i]－平均)×10/標準偏差 ＋ 50

という式で計算することにしましょう（この式は添字iの得点の偏差値）。ここで平均と標準偏差はどちらも科目nのものです。プログラム自体は前回に示したプログラムに対して，科目という添字が増えた分だけ，余分なforループを追加するだけです。

●ライフゲーム

ライフゲームはSX-WINDOWのサンプルプログラムとしても作ってみましたが，その核となる部分が二次元配列を用いたプログラムになっています。ライフゲームのプログラムをリスト2に示します（これはSX-WINDOW用とは異なっていますよ）。このライフゲームでは二次元配列の第1の添字をX座標，第2の添字をY座標と考えて，

char field[20][30]

という二次元配列を二次元のXY平面に対応させています。このとき，平面で座標が（x,y）の点は，

field[x][y]

という要素で参照でき，この値が1の場所は生存点，0の場所は非生存点ということになります。

なお，ライフゲームの変化の法則は次のようになっています。

1)　生存点は，まわりの生存点が2個か3個のときに生存できる。それ以外では消滅する。

2)　非生存点は，まわりの生存点が3個のときのみに生存点になる。

ある座標（x,y）のまわりの生存点の数は，

field[x−1][y−1]+field[x][y−1]
+field[x+1][y−1]+field[x−1][y]
+field[x+1][y]+field[x−1][y+1]
+field[x][y+1]+field[x+1][y+1]

によって計算できますから，この値によって次の世代（時刻）のfield[x][y]の値を0にするか1にするかをプログラムすればいいのです。なお，リスト2では生存点・非生存点の変化の状態を記憶するためにfieldと同じ大きさの二次元配列field1を使用していますが，勘のいい人ならば，この2つの配列をまとめて三次元配列として使用すればプログラムが少しすっきりするのがわかると思います。余力のある人は改造してみましょう。

●行列の計算

10元連立1次方程式を二次行列の操作によって解くプログラムがリスト3です。正しい解が求まったことを確認するためにこのプログラムには答え合わせをする部分も含まれています。

なお，リスト3ではなにやらごちゃごちゃやっていますが，これはガウス・ジョルダン法というアルゴリズムです。名称は物々しいですが，これは私たちが紙と鉛筆を使って方程式を解くのと同じことをやっているにすぎません。3元の連立方程式を解く単純な方法を思い出しましょう。

$ax+by+cz=p$　……（1）
$dx+ey+fz=q$　……（2）
$gx+hy+iz=r$　……（3）

という方程式がある場合，式（1）の両辺をaで割り，

$x+b'y+c'z=p'$　……（1）'

を作ったあと，

(2)－d×(1)'
(3)－g×(1)'

によって式(2)，(3)を，

$0x+e'y+f'z=q'$　……（2）'
$0x+h'y+i'z=r'$　……（3）'

変形します（xの係数を0にする）。これで式(1)'以外にxを含んだ式がなくなりました。次は(2)'の両辺をe'で割った式を何倍かして式(1)'，(3)'より引けば(2)'以外にyを含んだ式がなくなります。同様の操作を繰り返

せば，最終的に式(1)，(2)，(3)を，

x＋0y＋0z＝p''　…… (1)''

0x＋y＋0z＝q''　…… (2)''

0x＋0y＋z＝r''　…… (3)''

と変換することができて，これは方程式の解になっています。これが3元連立方程式の最も単純な解き方です。ガウス・ジョルダン法はこれと同じことを行列の操作だけで行っているだけなのです。

＊

前回の一次元配列と今回の二次元配列で配列の説明はおしまいです。配列はそのうちに説明するポインタと組み合わせて使うといろいろ面白いことができるのですが，配列だけしか知らなくてもC言語でかなりのプログラムが書けるはずです。

この連載で扱うサンプルプログラムも最近はだんだん複雑になってきているのがわかるでしょう。実用になるプログラム，というまでにはもう少し学ぶことがありますが，この調子でどんどん勉強していきましょう。さて，来月は文字列に関して説明してみたいと思います。それではまたお会いしましょう。

＊　＊　＊

◆基礎力を高めようの解答

設問1　1) 1　2) 2　3) 0　4) 5　5) 6　6) 7　7) 0　8) 6　9) 3

解説

静的な配列の初期化において初期値が与えられない要素は値が0になる（ANSI規格）。{　}で囲まれていない初期値は行の順序で並んでいる。この原則から与えられた配列の宣言は，

```
int x[3][4]={
        {1,2,0,0},
        {3,4,5,6},
        {7,8,9,0}  };
```

と読み変えられる。また[ ]は単なる演算子であるから添字が宣言した範囲に収まっているかのチェックはしない。添字が範囲外のときは配列要素が行の順序で並んでいることを思い出せばよい。

配列xの12個の要素は，

x[0][0], x[0][1], ……, x[0][10], x[0][11]

x[1][−4], x[1][−3], ……, x[1][6], x[1][7]

x[2][−8], x[2][−7], ……, x[2][2], x[2][3]

のいずれも参照できる。

リスト1

```
 1: /*
 2:          配列で統計的な処理を行う
 3:
 4:    配列markに格納されている得点の平均と標準偏差を求め、各自の偏差値を計算する
 5: */
 6: /* 平方根を求める関数 sqrt を使うためのオマジナイ */
 7: #include <math.h>
 8:
 9: int mark[3][15]={
10:  {60,75,80,90,88,51,15,30,60,81,77,68,65,93,88},
11:  {100,90,89,75,82,75,80,56,73,85,100,87,72,88,59},
12:  {55,80,71,60,65,90,85,94,98,77,67,80,48,76,58}
13: };
14: int v[3][15]; /* 偏差値を入れる */
15: double heikin[3],  /* 平均     */
16:        hyojun[3],  /* 標準偏差 */
17:        bunsan[3],  /* 分散     */
18:        sum[3];     /* 合計     */
19: int    n=15;       /* 人数  */
20: int    m=3;        /* 科目数 */
21:
22: main()
23: {
24:       int i,j;
25:       sum[0]=0;sum[1]=0;sum[2]=0;
26:       for(j=0;j<m;j++)
27:          for(i=0;i<n;i++)
28:             sum[j] = sum[j] + mark[j][i];
29:       for(j=0;j<m;j++)
30:          heikin[j] = sum[j]/n; /* 平均を求める */
31:       sum[0]=0;sum[1]=0;sum[2]=0;
32:       for(j=0;j<m;j++)
33:          for(i=0;i<n;i++)
34:            sum[j] = sum[j] + (mark[j][i]-heikin[j])*(mark[j][i]-heikin[j]);
35:       for(j=0;j<m;j++)
36:          bunsan[j] = sum[j]/n;              /* 分散というやつ */
37:       for(j=0;j<m;j++)
38:          hyojun[j] = sqrt(bunsan[j]);   /* これが標準偏差 */
39:       for(j=0;j<m;j++)
40:          for(i=0;i<n;i++)                /* 偏差値の計算です */
41:             v[j][i] = (mark[j][i]-heikin[j])*10/hyojun[j] + 50;
42:       for(j=0;j<m;j++){
43:          for(i=0;i<n;i++)
44:             printf("科目 %d, 番号 %d¥t: 得点 %d¥t: 偏差値 %d¥n",
45:                       j,       i,     mark[j][i],      v[j][i] );
46:          printf("科目 %d, 平均 %f : 標準偏差 %f¥n¥n", j, heikin[j], hyojun[j] );
47:       }
48: }
```

実行結果

```
科目 0, 番号 0  : 得点 60        : 偏差値 46
科目 0, 番号 1  : 得点 75        : 偏差値 53
科目 0, 番号 2  : 得点 80        : 偏差値 55
科目 0, 番号 3  : 得点 90        : 偏差値 60
科目 0, 番号 4  : 得点 88        : 偏差値 59
科目 0, 番号 5  : 得点 51        : 偏差値 42
科目 0, 番号 6  : 得点 15        : 偏差値 25
科目 0, 番号 7  : 得点 30        : 偏差値 32
科目 0, 番号 8  : 得点 60        : 偏差値 46
科目 0, 番号 9  : 得点 81        : 偏差値 55
科目 0, 番号 10 : 得点 77        : 偏差値 54
科目 0, 番号 11 : 得点 68        : 偏差値 49
科目 0, 番号 12 : 得点 65        : 偏差値 48
科目 0, 番号 13 : 得点 93        : 偏差値 61
科目 0, 番号 14 : 得点 88        : 偏差値 59
科目 0, 平均 68.066666 : 標準偏差 21.650147

科目 1, 番号 0  : 得点 100       : 偏差値 65
科目 1, 番号 1  : 得点 90        : 偏差値 57
科目 1, 番号 2  : 得点 89        : 偏差値 56
科目 1, 番号 3  : 得点 75        : 偏差値 45
科目 1, 番号 4  : 得点 82        : 偏差値 51
科目 1, 番号 5  : 得点 75        : 偏差値 45
科目 1, 番号 6  : 得点 80        : 偏差値 49
科目 1, 番号 7  : 得点 56        : 偏差値 30
科目 1, 番号 8  : 得点 73        : 偏差値 43
科目 1, 番号 9  : 得点 85        : 偏差値 53
科目 1, 番号 10 : 得点 100       : 偏差値 65
科目 1, 番号 11 : 得点 87        : 偏差値 55
科目 1, 番号 12 : 得点 72        : 偏差値 42
科目 1, 番号 13 : 得点 88        : 偏差値 55
科目 1, 番号 14 : 得点 59        : 偏差値 32
科目 1, 平均 80.733333 : 標準偏差 12.369137

科目 2, 番号 0  : 得点 55        : 偏差値 36
科目 2, 番号 1  : 得点 80        : 偏差値 54
科目 2, 番号 2  : 得点 71        : 偏差値 48
科目 2, 番号 3  : 得点 60        : 偏差値 40
科目 2, 番号 4  : 得点 65        : 偏差値 43
科目 2, 番号 5  : 得点 90        : 偏差値 61
科目 2, 番号 6  : 得点 85        : 偏差値 57
科目 2, 番号 7  : 得点 94        : 偏差値 64
科目 2, 番号 8  : 得点 98        : 偏差値 67
科目 2, 番号 9  : 得点 77        : 偏差値 52
科目 2, 番号 10 : 得点 67        : 偏差値 45
科目 2, 番号 11 : 得点 80        : 偏差値 54
科目 2, 番号 12 : 得点 48        : 偏差値 32
科目 2, 番号 13 : 得点 76        : 偏差値 51
科目 2, 番号 14 : 得点 58        : 偏差値 39
科目 2, 平均 73.600000 : 標準偏差 14.291256
```

リスト2

```
1: /*
2:          ライフゲーム（超手抜き版）
3: */
4: char field[15][20]={ /* 初期パターン（8の字） */
5:
6:     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
7:     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
8:     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
 9:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
10:    0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,
11:    0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,
12:    0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,
13:    0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,
14:    0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,
15:    0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,
16:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
17:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
18:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
19:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
20:    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
21:
22: };
23:
24: char field1[15][20]; /* パターンの一時記憶用 */
25:
26: main()
27: {
28:         int gen;        /* 世代（時間）*/
29:         int goukei;       /* 生存点の合計 */
30:         int x,y,t;
31:
32:         printf("何世代先?");
33:         scanf("%d", &gen);
34:
35:         for(t=0;t<=gen;t=t+1){
36:
37:            printf("世代=%d¥n", t ); /* 世代ごとにプリント */
38:            for(x=0;x<15;x=x+1){
39:               for(y=0;y<20;y=y+1){
40:                  if(field[x][y]==1) printf("●");
41:                  else               printf("  ");
42:               }
43:               printf("¥n");
44:            }
45:
46:            for(x=1;x<14;x=x+1) /* 平面の端っ子は見ない */
47:               for(y=1;y<19;y=y+1){
48:                  goukei=
49:                      field[x-1][y-1] + field[x][y-1]
50:                     +field[x+1][y-1] + field[x-1][y]
51:                     +field[x+1][y]   + field[x-1][y+1]
52:                     +field[x][y+1]   + field[x+1][y+1];
53:                  if(field[x][y]==1){ /* 生存点 */
54:                     if(goukei==2 || goukei==3)
55:                        field1[x][y]=1;
56:                     else
57:                        field1[x][y]=0;
58:                  } else {            /* 非生存点 */
59:                     if(goukei==3)
60:                        field1[x][y]=1;
61:                     else
62:                        field1[x][y]=0;
63:                  }
64:               }
65:
66:            for(x=1;x<14;x=x+1) /* 一時記憶を戻す */
67:               for(y=1;y<19;y=y+1)
68:                  field[x][y]=field1[x][y];
69:
70:         }
71: }
```

実行結果

```
世代=0

●●●
●●●
●●●
　　　●●●
　　　●●●
　　　●●●

世代=1

　　●
　●　●
●　　　●
　●　　　●
　　●　　　●
　　　●　　　●
　　　　●　●
　　　　　●

世代=2

　　●
　●●●
●●●　●
　●　　　●
　　●　　　●
　　　●　●●●
　　　　●●●
　　　　　●

世代=3

　●●●
●
●　　　●
●　　●　●
　　●　●　　●
　　　●　　　●
　　　　　　　●
　　　　●●●
```

リスト3

```
 1: /*
 2:
 3: ガウス・ジョウルダン法で連立一次方程式を解く
 4:
 5: A[0][0]·X0 + A[0][1]·X1 + … + A[0][N-1]·XN-1 = A[0][N]
 6: A[1][0]·X0 + A[1][1]·X1 + … + A[1][N-1]·XN-1 = A[1][N]
 7: ……
 8: A[N-1][0]·X0 + A[N-1][1]·X1 + … + A[N-1][N-1]·XN-1 = A[N-1][N]
 9:
10: という形式の方程式を仮定する（配列宣言は A[N][N+1] ）。
11: ただし、簡単のために A[i][i]≠0 (i=0,…,(N-1))とする。
12:
13: */
14: double A[10][11]={ /* N=10 */
15:    1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,
16:    1,  4,  1,  4,  2,  1,  3,  5,  6,  2,  3,
17:    3,  1,  4,  1,  5,  9,  2,  6,  5,  3,  5,
18:    2,  7,  1,  8,  2,  8,  1,  8,  2,  8,  4,
19:    3,  5,  9,  5,  3,  8,  6,  2,  6,  9,  7,
20:    1,  3,  5,  7,  9, 11, 13, 17, 19, 23, 29,
21:   31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
22:    1,  1,  2,  2,  3,  3,  4,  4,  5,  5,  6,
23:    2,  2,  2,  2,  2,  2,  2,  2,  2,  2,  2,
24:    2,  2,  3,  6,  0,  6,  7,  9,  1,  2,  3,
25: };
26:
27: double B[10][11]; /* 答え合わせ用の控え */
28:
29: main()
30: {
31:         int N=10;
32:         int i,j,k;
33:         double tmp;
34:
35:         for(i=0;i<N;i=i+1) /* AをBにコピー */
36:            for(j=0;j<(N+1);j=j+1) B[i][j]=A[i][j];
37:
38:         for(i=0;i<N;i=i+1){
39:
40:            for(j=(i+1);j<(N+1);j=j+1) /* 対角成分を1に */
41:               A[i][j]=A[i][j]/A[i][i];
42:
43:            for(j=0;j<N;j=j+1){ /* 対角成分以外を0にする */
44:               if(i==j) continue;
45:               for(k=(i+1);k<(N+1);k=k+1)
46:                  A[j][k]=A[j][k]-A[j][i]*A[i][k];
47:            }
48:
49:         }
50:
51:         printf("¥n解¥n");
52:         for(i=0;i<N;i=i+1) /* 解をプリント */
53:            printf("(%d) %f¥n",i,A[i][10]);
54:
55:         printf("¥n答え合わせ¥n");
56:         for(i=0;i<N;i=i+1){
57:            tmp=0;
58:            for(j=0;j<N;j=j+1) /* 方程式の値を計算 */
59:               tmp=tmp+B[i][j]*A[j][10];
60:            printf("(%d) %f は %f ですか?¥n",i,tmp,B[i][10]);
61:         }
62: }
```

実行結果

```
解
(0) 4.301768
(1) -0.268572
(2) -2.579361
(3) -1.500421
(4) -3.709017
(5) 1.167462
(6) 0.847933
(7) -0.658261
(8) 2.138676
(9) 1.259792

答え合わせ
(0) 11.000000 は 11.000000 ですか?
(1) 3.000000 は 3.000000 ですか?
(2) 5.000000 は 5.000000 ですか?
(3) 4.000000 は 4.000000 ですか?
(4) 7.000000 は 7.000000 ですか?
(5) 29.000000 は 29.000000 ですか?
(6) 30.000000 は 30.000000 ですか?
(7) 6.000000 は 6.000000 ですか?
(8) 2.000000 は 2.000000 ですか?
(9) 3.000000 は 3.000000 ですか?
```

PurePASCAL

PASCALプログラミングへの招待〈最終回〉

Garbage Collection in Pure PASCAL

PurePASCAL講座も今回が最終回。そこで，これまでの基本文法解説で説明の不十分だったものを集めてまとめて解説してみました。連載は終了しますが，これからもPurePASCALを愛用してみてください。

Fujiki Takeshi / Fujii Yoshimi
藤木健士/藤井義巳

連載は今回が最終回となりますので，今回はこれまでに説明し残した部分のチリ集め（ガーベジコレクション）をするとしましょう。タイトルを見て，「おや？PASCALもガーベジコレクションをするのかな？」と思われた方もいるかと思いますが，全然関係ありません。

可変部付きレコード型

型の説明のところで（昨年8月号）「(レコード型の)可変部とは，C言語でいう共用体にあたるものです」と書いておきながら，それ以上の説明はなにもしていないことに気がつきましたので，ここで説明します。レコード型の構文定義をもう一度書きますと，

```
type  レコード型名＝record
      固定部
      可変部
end；
```

といったかたちをしています。固定部，可変部はそれぞれ省略できます。8月号ではComplex（複素数）型を例に，可変部のないレコード型の説明をしました。Complex型は次のようなかたちをしていましたね。

```
type Complex＝record
    Re, Im : Real
end ;
```

これをC言語で表現すると，次のようになります。

```
typedef struct Complex {
        double Re, Im ;
    } Complex ;
```

可変部のあるレコード型宣言の構文は少々複雑です。次の例は名簿を作るときのデータ構造の例を示しています。レコード型Personは固定部に3つのフィールド変数name, age, sを持っています。また，可変部にはCountry, Homeの2つのフィールド変数がありますが，この2つは同時にアクセスすることはできません。この2つのどちらをアクセスするかを決めるのが選択子foreignerです。選択子の型は任意の順序型を用います。可変部のフィールドにアクセスするためにはまずこの選択子foreignerにTrueかFalseの値を代入します。Trueを代入したならCountryが，Falseを代入したならHomeがアクセスできます。

```
type Person＝record
    name : Str255 ;
    age : Integer ;
    s : Sex ;
    case foreigner:Boolean of
        True : (Country : Str255) ;
        False : (Home : Str255)
end ;
```

同じデータ構造をC言語で表現すると，次のようになります。

```
typedef struct Person {
        char name[256];
        int age ;
        Sex s ;
        Boolean foreigner;
        union {
            char Country[256];
            char Home[256];
        } d;
    } Person ;
```

例があまりよくなくて，共用体のメンバーが同じ型なのですが，説明のためですので我慢してください。共用体のメンバーのうちどちらを使うかを決定する情報はC言語では共用体には含まれません。また，共用体にも名前をつけなければいけません。Cらしいといえばいえるのですが，共用体を単独で使用するケースはほとんどなく，構造体の中に含まれるのが普通です。そういった場合，PASCALならPerson型の変数pのフィールドCountryにアクセスするとき，

```
write(p.Country) ;
```

と書けば済むところをC言語では，

```
printf("%s", p.d.Country);
```

と書かなければいけません。PASCALからの移植で共用体が2重，3重になると目もあてられません。

共用体の名前を長くすると書くのが大変なので，たいていの人は1文字か2文字の意味のない名前をつけるのではないでしょうか。そうするとソースがますます見にくくなります。喜ぶべきことにC＋＋では共用体に名前

をつける必要はなくなりました。

手続き引数と関数引数

プログラムの種類によっては手続きや関数そのものを引数としてほかの手続き，関数に渡したいことがあるでしょう。PASCALではそのようなことも可能です。しかし，これも可変部付きのレコード型に負けず劣らず複雑なので，初心者の方は使い方でかなり苦しむのではないかと思います。

手続き引数と関数引数の実引数（厳密な説明ではないが，手続き，関数呼出しのときに呼出し側が渡す引数をいう。被呼側から見た引数は仮引数という）には標準手続きや標準関数を用いることはできないなど，制限も多くあります。

手続き，関数引数の宣言は，手続き，関数の頭部そのままです。

```
procedure P1 (procedure P2 (i : integer) ;
function F1 (r : real) : real) ;
```

例の手続きP1は手続き引数P2と関数引数F1を持っています。P2は引数に整数型のｉを持ち，F1は引数に実数型のｒを持ちます。さらにF1は実数を返します。P1の中でP2とF1は普通の手続き，関数のように用いることができます。とまあこのようなものです。

サンプルプログラムを挙げて説明しようと思ったのですが，あまりいい例は思いつきませんでした。リストは私が数年前，まだPurePASCALを作り始めてもいなかった頃にレポートとして書いたプログラムを，関数引数を用いて書き直したものです。手続きDisplayがそれですが，関数引数がなければ同じようなルーチンを２回書くことになります。プログラムは数値積分の練習問題です。私はあまり得意ではありませんので解説は省略です。このあたりを詳しく知りたい方は共立出版の「数値計算の常識」でも読んでください。

Ｃ言語では手続き，関数引数に相当するものとして関数へのポインタがあります。

```
typedef void ( *func)( ) ;
```

このように宣言しておくと，func型の変数には関数を代入できるし，引数がfunc型ならば引数に関数を与えることもできます。もちろん構造体にfunc型のメンバーを含むことも可能です。PASCALの手続き，関数引数もこれくらい自由に使えれば，もう少し使い道があると思います。

コンパイラに興味がある人のために

1950年代の終わりに最初のFORTRANコンパイラが作られた頃はコンパイラはもっとも複雑で作成が困難なプログラムとされていました。その後，幾多の研究者の努力で形式言語理論に代表されるコンパイラの理論が確立されました。また，その理論にもとづいて有名なyacc, lexなどのツールも作られ，最近ではコンパイラ作成作業の多くの部分は自動的に行われるようになっています。ここ２，３年のあいだに日本でもコンパイラの教科書や，yacc，lex（注：X68000にもGNUプロジェクトのyaccとlexに相当するbisonとflexが移植されています）を使った実践的な解説書が多数出版され，私も含めてこの道を志す学生には嬉しい限りです。

なかでも先日ついに邦訳が発売されたAhoとUlmanの教科書，そのタイトルもズバリ「コンパイラ」（［１］［２］）は世界的なベストセラーの教科書です。

残念なのは邦訳の値段が２冊組で１万円を越え，原書のペーパーバックの４倍も高いことです。ですから英語の得意な方は原書をおすすめします。その他にもおすすめの本があるので列挙します。

［１］Aho,A.V.,Sethi,R. and Ulman,J.D.:"Compilers-Principles,
Techniques, and Tools-", Addison-Wesley, 1986

［２］原田賢一訳，「コンパイラⅠ・Ⅱ」，サイエンス社，1990　［１］の邦訳。

［３］佐々政孝，「プログラミング言語処理系」，岩波書店，1989

コンパイラの理論全般にわたる本格的な教科書。セマンティクスの説明に属性文法による記述を多用しているのが特徴。

［４］疋田輝雄・石畑清，「コンパイラの理論と実現」

リスト１

```
=================== LIST ===================
     1: (* 'sample.pas' *)
     2: (*
     3:            数値積分のレホートプログラム
     4:            Modified Dec.1990 by Chack'n
     5: *)
     6:
     7: program report2a(input,output);
     8:
     9:  const pai = 3.141592;
    10:
    11:  type fkind=(a1,a2,b1,b2,c1,c2,d,e);
    12:
    13:  var   funcno:fkind;
    14:        a,b,aa,bb:real;
    15:        func:array[fkind]of packed array[1..2] of char;
    16:        n:integer;
    17:        x:real;
    18:
    19: function f(x:real):real;
    20:   begin
    21:      case funcno of
    22:         a1: f:=sqrt(x);
    23:         a2: f:=sqrt(x);
    24:         b1: f:=x*x*exp(-x*x*x);
    25:         b2: f:=x*x*exp(-x*x*x);
    26:         c1: f:=1/(1+x*x);
    27:         c2: f:=1/(1+x*x);
    28:         d:  f:=sqrt(1-x*x);
    29:         e:  f:=x*exp(x)
    30:      end
    31:   end;
    32:
    33: function integralF(x:real):real;
    34:   begin
    35:      case funcno of
    36:         a1: integralF:=2*sqrt(x*x*x)/3;
    37:         a2: integralF:=2*sqrt(x*x*x)/3;
    38:         b1: integralF:=-exp(-x*x*x)/3;
    39:         b2: integralF:=-exp(-x*x*x)/3;
    40:         c1: integralF:=arctan(x);
    41:         c2: integralF:=arctan(x);
    42:         d:  integralF:=sin(2*x)/4-x/2;
    43:         e:  integralF:=(x-1)*exp(x)
    44:      end
    45:   end;
    46:
    47: procedure AandB;
```

```
 48:   begin
 49:      case funcno of
 50:         a1: begin a:=0;b:=1;aa:=0;bb:=1 end;
 51:         a2: begin a:=1;b:=2;aa:=1;bb:=2 end;
 52:         b1: begin a:=-1;b:=1;aa:=-1;bb:=1 end;
 53:         b2: begin a:=0;b:=1;aa:=0;bb:=1 end;
 54:         c1: begin a:=0;b:=1;aa:=0;bb:=1 end;
 55:         c2: begin a:=1;b:=2;aa:=1;bb:=2 end;
 56:         d:  begin a:=-1;b:=1;aa:=pai;bb:=0 end;
 57:         e:  begin a:=0;b:=1;aa:=0;bb:=1 end
 58:      end
 59:    end;
 60:
 61: function daikei(a,b:real;n:integer):real;
 62:  var h,val:real;
 63:      i:integer;
 64:   begin
 65:      h:=(b-a)/n;
 66:      val:=f(a)+f(b);
 67:      for i:=1 to n-1 do
 68:         val:=val+2*f(a+i*h);
 69:      daikei:=h*val/2
 70:   end;
 71:
 72: function simpson(a,b:real;n:integer):real;
 73:  var h,val:real;
 74:      i:integer;
 75:   begin
 76:      h:=(b-a)/n;
 77:      val:=f(a)+f(b);
 78:      for i:=1 to (n div 2) do
 79:         val:=val+4*f(a+(2*i-1)*h);
 80:      for i:=1 to (n div 2 - 1) do
 81:         val:=val+2*f(a+2*i*h);
 82:      simpson:=val*h/3
 83:   end;
 84:
 85: function sinchi(a,b:real):real;
 86:   begin
 87:      sinchi:=integralF(b)-integralF(a)
 88:   end;
 89:
 90: procedure init;
 91:   begin
 92:      func[a1]:='a1';
 93:      func[a2]:='a2';
 94:      func[b1]:='b1';
 95:      func[b2]:='b2';
 96:      func[c1]:='c1';
 97:      func[c2]:='c2';
 98:      func[d]:='d ';
 99:      func[e]:='e '
100:   end;
101:
102: (* 関数引数を持つ手続きの例 *)
103:   procedure Display(n:integer; a, b:real;
104:                 function AI(a,b:real;n:integer):real);
105:   var
106:     x:real;
107:   begin
108:      write(n:6);
109:      x:=AI(a,b,n);
110:      write(x:20);
111:      writeln(abs(x-sinchi(aa,bb)):20);
112:   end;
113:
114: { main routine }
115:   begin
116:      for funcno:=a1 to e do
117:         begin
118:            AandB;
119:            init;
120:            writeln('<function:',func[funcno],'> ');
121:            writeln(' - daikei method -');
122:            writeln('[n] ':6,'[daikei]      ':20,'[gosa]      ':20);
123:            n:=2;
124:            while n<=64 do
125:               begin
126:                  Display(n, a, b, daikei);    (* ここで使われる *)
127:                  n:=n*2
128:               end;
129:            writeln;
130:            writeln(' - simpson method -');
131:            writeln('[n] ':6,'[simpson]     ':20,'[gosa]      ':20);
132:            n:=2;
133:            while n<=64 do
134:               begin
135:                  Display(n, a, b, simpson);   (* ここでも使われる *)
136:                  n:=n*2;
137:               end;
138:            writeln
139:         end
140:   end.
```


Cのサブセットのコンパイラを例に，コンパイラの理論と実現のテクニックを解説している。

［5］Pemberton, S., Daniels, M., 邦訳　武市正人，木村友則，「Pascalの言語処理系 Pascal-P4」，近代科学社，1984

PASCAL自身で書かれたPASCALコンパイラの全ソースリストとその解説。実際にコンパイラのコーディングをする際，データ構造などは大変参考になる。Pコードの説明も載っている。

［6］「Cコンパイラ設計」

yaccとlexを使ってCのサブセットのコンパイラを作る。yaccの解説書として最適。

［7］近藤嘉雪，「yaccによるCコンパイラプログラミング」，ソフトバンク，1990

ご存じCマガジンの連載記事を本にしたもの。題名どおりyaccを使ってかなり実践的にコンパイラの作り方を説明している。

最後に

半年以上にわたって私たちの連載を読んでいただきまして，ありがとうございます。C言語全盛の今日，いまさらPASCALなんてと思われた方もなかにはいらっしゃるでしょう。しかし，PASCALにはCにない特徴がいくつもあります。PASCALで学んだことは，Cでプログラムを組む際にもきっと役に立つことでしょう。

また，読者の皆さんのなかには将来オリジナルの言語を作ってやろうと思っている人もいると思いますが，そのときその言語の土台としてPASCALは良質の素材になります。Modula-2, Ada, Concurrent PASCALなど，PASCALをベースに設計された言語は数えるときりがないほどたくさんあります。プログラミング言語の世界では現在，オブジェクト指向が大流行です。Turbo PASCALはバージョン5.5からオブジェクト指向PASCALになりました。パソコン用のC++の処理系もそろそろ出揃ってきたようです。

我らがX68000にもユーザーの手でG++(GCCのオブジェクト指向版？　C++とは異なる）が移植されています。私もCには限界を感じていますので，時期主力言語としてC++には大いに期待しています。読者の皆さん，C++の勉強は始めていますか？

この連載を通じてPASCALの言語仕様とコンパイラに関することを説明したわけですが，これを読んでプログラミング言語や言語処理系に関して興味を持っていただけたのであれば幸いです。皆さんも言語処理（コンパイラや自然言語処理）などのプログラムを作ってみるのどうですか？　そうでないプログラムでもちょっと大きめのプログラムを書いてみるのはどうでしょうか。動作したときの喜びも大きいものです。これからもユーザーの側からX68000をもりあげていこうではありませんか！

最後にPurePASCALの作成に際していろいろ教えてくださった九州大学工学部情報工学科の荒木啓二郎先生には深く感謝致します。また多大な励ましをいただいた日本IBMの鈴木宏治さんにも感謝いたします。

ソーティングプログラム(後編)

Murata Toshiyuki 村田 敏幸

アルゴリズムの面白さと重要性を実感できるソーティングの技法の数々。今回はその後編で，より高速なソーティング法を紹介します。いずれの場合でも，アルゴリズムの特徴を理解して効果的にプログラムを作成することの大切さがおわかりいただけることでしょう。

ソーティングの後編である。前編では，比較的単純で直感的な3種類のアルゴリズムを取り上げ，まずはアルゴリズムどおりにプログラムを書いておき，ついで実行速度を追求すべく，できる限りの最適化を施した[1]。今月は“あのような最適化”をしたことが空しくなるほど，圧倒的に高速なソーティングアルゴリズムを紹介する。

なお，関連知識として，コラムで“計算量”という概念についてまとめておく。

シェルソート

シェルソート（Shell sort）は単純挿入法の改良案だ。まず離れた要素間の順序を大雑把に整え，徐々に近くの要素間の順序を整えていく。処理時間が要素数nの2乗に比例する単純法に対し，シェルソートはだいたい$n^{1.5}$に比例する程度の処理時間でソートをこなす。ちなみにシェルはアルゴリズムの考案者の名前である。

●アルゴリズム

1) 添え字の適当な“跳び幅”を設定し，配列中の要素をその跳び幅ごとに拾って“群”を作る（と考える。実際にデータを動かすわけではない)。跳び幅をgとすれば，配列はg個の群に分割される。
2) 各群をそれぞれ単純挿入法でソートする。
3) 跳び幅を適当に減らして，1)，2)を繰り返す。
4) 跳び幅1で1)，2)を行った時点でソートは完了する。

この手順で，どうしてソートができるのか悩んではいけない。最終的には跳び幅1で分割して(＝分割

1) そのつもりだったのだが，甘かった。先月最後のプログラム，リスト8は，まだリスト1のように最適化できる。

リスト1　SORT6.S修正版
（先月のリスト8からの変更点のみ）

```
 3: *           8000要素のソートに 74.10秒
22:             bra      next1
23:
24: shift:      move.l   d1,-8(a2)
25:
26: loop1:
27: loop2:      move.l   (a2)+,d1
28: next2:      cmp.l    d1,d0
29:             bgt      shift
30:
31:
32: found:      move.l   d0,-8(a2)
```

リスト2　SORT7.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (シェルソート)
 3: *           8000要素のソートに 1.65秒
 4: *
 5:             .xdef    sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l    1              *配列の先頭アドレス
 9: ABOT:       .ds.l    1              *配列の最終アドレス+1
10: *
11:             .text
12:             .even
13: *
14: sort:
15:             link     a6,#0
16:             movem.l d0-d3/a0-a4,-(sp)
17:
18:             movem.l ATOP(a6),a0-a1  *a0=配列先頭アドレス
19:                                     *a1=配列末尾+1アドレス
20:             moveq.l #$ffff_fffc,d3  *d3=下位2ビットマスク用データ
21:             move.l   a1,d2          *
22:             sub.l    a0,d2          *d2=配列のバイト数
23:             bra      next1
24:
25: shift:      move.l   d1,(a4)        *のちの挿入に備えて
26:                                     *データをずらす
27: loop1:
28: loop2:
29: loop3:      movea.l a3,a4           *a4=挿入予定位置
30:             suba.l   d2,a3          *a3=検索位置
31:             cmpa.l   a0,a3          *配列の先頭に達したら
32:             bcs      insert         *  群の先頭に挿入する
33: next3:      move.l   (a3),d1        *
34:             cmp.l    d1,d0          *比較を繰り返して
35:             blt      shift          *  挿入位置を探す
36:
37: insert:     move.l   d0,(a4)        *挿入する
38:
39: next2:      movea.l a2,a3           *a3=挿入予定位置
40:             move.l   (a2)+,d0       *d0=挿入するデータ
41:             cmpa.l   a1,a3          *未整理要素がなくなるまで
42:             bcs      loop2          *  繰り返す
43:
44: next1:      lsr.l    #1,d2          *跳び幅を半減する
45:             and.l    d3,d2          *
46:             lea.l    0(a0,d2.l),a2  *a2=整理済み部分と
47:                                     *  未整理部分の境界
48:                                     *(頭のG個は整理済み)
49:             bne      next2          *跳び幅が0になるまで繰り返す
50:                                     *(leaではccrは変化しない)
51:
52: done:       movem.l (sp)+,d0-d3/a0-a4
53:             unlk     a6
54:             rts
55:
56:             .end
```

しないで）単純挿入法を適用するのだから，ソートできることは間違いないのだ。問題は，なぜ，分割して何回もソートすると，まとめて一度でソートするよりも速くなるかだが，これは次のように考えよう。

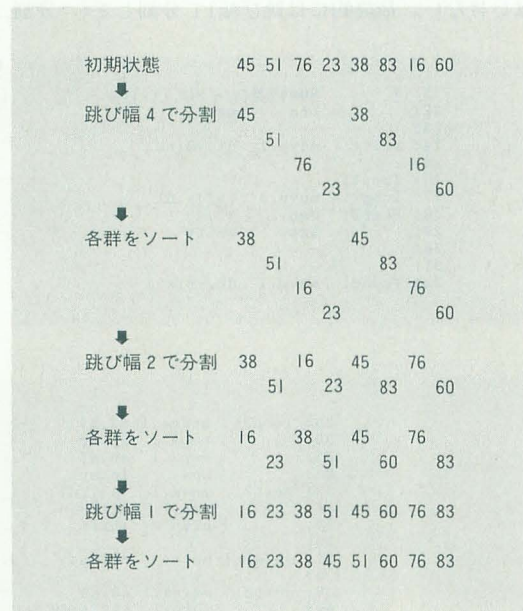
単純挿入法によるソート時間はn^2に比例する。要素数が2倍になれば処理時間は4倍になり，3倍になれば9倍になる。逆にいうと，要素数が半分になれば処理時間は1/2，1/3になれば1/9，というように，要素数が減るにつれ，処理時間は急激に短くなっていく。これが前提1。また，単純挿入法では，データの初期並びが整順に近ければ近いほどデータの移動回数が少なくてすむ。極端な場合，配列がすでにソートずみであれば，データの移動は一切ない。これが，前提2。

シェルソートの序盤では，各群に含まれる要素数は非常に少ない。前提1より，各群をソートするのにかかる時間は，全体をソートする時間に比べればごくわずかだといってよい。また，終盤では，各群は長くなるものの，比較的整順に近づいている。前提2より，やはりソート時間は短くてすむことが想像できる。この組み合わせがシェルソートの速度の秘密というわけだ。

図1に8個の要素を跳び幅4，2，1でシェルソートする様子を示しておく。

2）とかいいながら，リスト2では効率を優先して“配列の直後のメモリを読み出す（書き込みはしない）”という反則技を前回に続いて使っていたりする。マシン語プログラマやCプログラマはこういった点にはルーズなものだが，PASCAL派には怒られるかもしれない。

図1　シェルソート

初期状態	45	51	76	23	38	83	16	60
跳び幅4で分割	45				38			
		51				83		
			76				16	
				23				60
各群をソート	38				45			
		51				83		
			16				76	
				23				60
跳び幅2で分割	38		16		45		76	
		51		23		83		60
各群をソート	16		38		45		76	
		23		51		60		83
跳び幅1で分割	16	23	38	51	45	60	76	83
各群をソート	16	23	38	45	51	60	76	83

リスト3　SORT8.S（リスト2からの追加・変更部）

```
20:            moveq.l #$ffff_fffc,d3   *d3=下位2ビットマスク用データ
21:            move.l  a1,d0            *
22:            sub.l   a0,d0            *d0=配列のバイト数
23: *          lsr.l   #1,d0
24:            moveq.l #4,d2            *配列のデータ数に
25: loop0:     add.l   d2,d2            *  最も近い
26:            addq.l  #4,d2            *  2^x-1の数を求め
27:            cmp.l   d0,d2            *  跳び幅の初期値とする
28:            bcs     loop0            *
29:            bra     next1
```

●プログラム例

リスト2は跳び幅gの初期値としてn/2を使い，以下，パス1回ごとにgを半減していく（端数切り捨て）という，よく見かけるタイプのシェルソートルーチンだ。真正直にアルゴリズムを実現しようとすると，単純挿入法の2重ループを，各群に対して繰り返すループと，跳び幅が1になるまで繰り返すループでくくった4重ループになってしまいそうだが，リスト2では各群のソートを並行して行うことでいくらかシンプルな3重ループで抑えている。

跳び幅はd2レジスタに収め，44行の，

```
lsr.l   #1, d2
```

によって半減していく。ただし，d2には添え字レベルの跳び幅ではなく，配列の要素サイズ（いま扱うのは4バイトデータだから4）を掛けたアドレスレベルでの跳び幅を格納する形にした。こうしておけば，ある要素を指しているポインタに対してd2を加減算するだけで，同一群中の前後の要素のアドレスが求まり，わざわざ添え字からアドレスを計算するといった手間を省くことができる。その代わり，d2はつねに4の倍数に保たなければならず，

```
lsr.l   #1, d2
```

によって2で割ったときには，さらに下位2ビットをマスクする処理が必要となる（45行）。

●改良案

先月，単純挿入法に番人を導入することで高速化する例を示した。シェルソートでも，各群のソートには単純挿入法を使うわけだから，同様の手法で高速化が図れる。だが，各群ごとに番人を置く必要があるため，番人のためだけにかなり余計なメモリが必要になる。

リスト2では跳び幅gの初期値は配列の要素数nの半分であり，番人を置くためだけに配列のバイト数/2バイトのメモリが要求される計算だ。gの最大値がなるべく小さくなるように細工すれば，番人の個数も少なくてすむとはいえ，先月も触れたように，番人を置くアルゴリズムはメインルーチン側に少なからず負担をかけるし，そもそも余分なメモリを使うというのは僕の美意識に反する2)。というわけで，今回は番人を置くアプローチは諦め，別の方法を探ってみた。

シェルソートの効率はgの減らし方の系列をどのようにとるかによって変わってくる。リスト2に示したタイプは，gを求めるのが簡単な割には効率も悪くはないのだが，gが常に奇数になるよう初期値を工夫すると，もう少し速くなることが知られている。試しにリスト2の20～23行をリスト3で置き換えてみよう。いくらか，ソート速度が向上する。

リスト3のループでは，跳び幅の初期値として，

1, 3, 7, 15, 31, 63, ……[3)]

のような2^x-1の数のうち，内輪でもっともnに近い値を求めている（ただし，すでに述べたような理由で4倍してある）。余計な処理を入れると実行速度が低下しそうな気がするが，このループの計算量は$\log_2 n$のオーダーにすぎず，nがある程度より大きければ大勢には影響しない。

ところで，リスト3のループで求まるgの初期値はn/2以上n未満であり，最初のパスでは，要素が1個しかない群が生じる。要素が1個の群をソートするのは明らかな無駄だ。この無駄を省くには，gの初期値をもう1ステップ小さくし，各群に最低2個の要素がある状態から始めるようにすればよい。注釈で殺してある23行を復活してみてほしい。この変更により，gの初期値はn/4以上n/2未満（各群に最低でも2要素）となる。

さて，ほんの気紛れで，gの初期値をさらに小さくしてみたら，予想外によい結果が出た。23行を，

```
lsr.l  #2, d0
lsr.l  #3, d0
lsr.l  #4, d0
lsr.l  #5, d0
```

などに変更して実行時間を比較してみてもらいたい。与えるデータによっても異なるかもしれないが，試してみた範囲では，

```
lsr.l  #3, d0
```

を選択した場合，つまり，gの初期値がn/16以上n/8未満（各群に最低でも8要素）のときに最高速になり，それ以上小さくすると急激に遅くなるという結果が出た。

3) gとしてこのような数列（の逆順）を使った場合，計算量は$n^{1.2}$のオーダーとなる。

ヒープソート

ヒープソート（heap sort）は，ある意味で単純選択法の改良案といえる。単純選択法同様，1回のパスごとに最小値を1つずつ選んでいくのだが，その過程で，以降の選択が楽に行えるような情報を残す。また，ソートするデータの初期並びにはほとんど影響されず，コンスタントによい性能を発揮するのもヒープソートの特徴だ。

●アルゴリズム

1) まず，ソートすべきデータ列をヒープに再構成する。ここでいうヒープとは，たとえば図2のようなものだ。いわゆる2分木（binary tree：2進木ともい

コラム　計算量

アルゴリズムの速度は，時間計算量（time complexity），あるいは，単に計算量（complexity）と呼ばれる尺度で計られる。詳しくは『The Art of Computer Programming』のあたりを読んでもらうとして，ここではごく簡単に説明しておこう。

計算量は，プログラムの記述言語や走行環境（プロセッサの種類やクロック）といった要素を除外した，アルゴリズムの純粋な速度を示すものだ。だが，それだからこそ，厳密な数値で表現できるものではなく，計算量は“処理するデータの個数nが増えると実行速度がどう変わるか”，“実行時間がどのような関数f(n)に比例するか”，という程度の精度で表される。

たとえば，nが2倍になると，実行時間が4倍になる（実行時間がn^2に比例する）とき，そのアルゴリズムの計算量はn^2のオーダーである，というういいかたをする。nが増えたときのf(n)の増加傾向がなだらかであればあるほど，性能のよいアルゴリズムとみなすわけだ。計算量が$\log_2 n$のオーダーのアルゴリズムと，nのオーダーのアルゴリズムがあったとすると，$f(n)=\log_2 n$は$f(n)=n$に比べればずっとなだらかな増加傾向を示すことから，前者のほうが優れている，と考える（老婆心ながら，$\log_2 n$は“2を底とした対数”で，$2^x=n$のときのxの値を意味する。$\log_2 2=1$，$\log_2 4=2$，$\log_2 8=3$という具合で，要は“nを2進数で表したときの桁数−1”だ）。

“f(n)に比例する”なんていう大雑把な表現で，アルゴリズムの性能を判断することに疑問を感じる人もいるかもしれない。確かに，このような形で表された計算量はnが十分大きいときの平均的な性能を示しているにすぎず，nが非常に小さいときには，表には出てこない微小要素（たとえば，初期化にかかる時間）も無視できなくなるし，比例定数の大小も大きく影響してくる。上の例で，仮に$\log_2 n$にかかる比例定数のほうが数倍大きかったとすると，nが小さいときに限り，オーダーnのアルゴリズムのほうが速くなる。

とはいっても，nが非常に小さな場合というのはいわば例外だ。一般には，大量のデータを高速に処理できるアルゴリズムのほうが重宝であることを考えると，計算量という尺度は“アルゴリズムの性能を計る目安”として十分な精度と意味を持つといえるだろう。

では，試しに先月のバブルソートアルゴリズム（なんの工夫もしない版）の計算量を計算してみよう。ソーティングはデータの比較と移動という2つの基本操作で成り立つわけだから，それぞれの実行回数がnの関数で表せれば，計算量が求まる。

バブルソートの各パスにおける比較回数は，最初のパスではn−1回，2回目のパスではn−2回，以下，1ずつ減っていき，最後のパスでは1回だけ比較が行われる。比較の総回数は1からn−1までの和であり，

$$n(n-1)/2$$
$$=n^2/2-n/2\text{回}$$

となる。ここで，nが十分大きいときには，第2項は第1項に比べて無視できるほど小さくなると考えられる。また，いまは正確な比較回数を求めたいのではなく，比較回数がnのどのような関数に比例するかがわかれば十分であり，1/2という定数係数は気にする必要がない。結局，この式からは，バブルソートの比較回数がn^2のオーダーであるという情報が得られる。

続いて，データ移動の回数だ。移動回数は，ソートしようとするデータ列の並びに影響を受けるから一意には求まらないが，仮に比較2回につき1回交換が行われるとみなし，また，1回の交換を，

```
temp = A
A = B
B = temp
```

の3回のデータ移動に置き換えると，平均，

$$3n(n-1)/4$$
$$=3n^2/4-3n/4\text{回}$$

のデータ移動が起こると考えられる。比較回数のときと同様の考え方で，データ移動の回数もn^2のオーダーだということがわかる。

データの比較回数も移動回数もn^2のオーダーであることから，バブルソートの計算量はn^2のオーダーだという結論に達する。この結果は，先月の表1に示した実測値とも一致する（あの表は，単純××法の計算量がどれもn^2のオーダーであることを暗示している）。

上では，比較と移動の総回数から計算量を求めたわけだが，各パスごとの比較回数・移動回数と，パス回数からも同様の結果が導き出せる。各パスごとの比較回数・移動回数はどちらも，

$$an+b\ \ (a,\ b\text{は定数})$$

のようなnの1次式で表されるから，nのオーダーであり，パス回数はn−1で，やはりnのオーダーだ。かけ合わせれば，全体でn^2のオーダーとなる。

う）だが，“それぞれの節（node）に置かれたキーは子＝両方の枝（branch）にぶら下がった下位節に位置するキーよりも大きい（か，等しい）”という条件を満たしている。

2）　ヒープの定義より，根(root：図2でいう一番上の節。データ構造の“木（tree）”は根が上にあり，下向きに枝を伸ばすように描かれる）にはもっとも大きな要素がある。これを取り出し，木のなるべく下位レベルの，なるべく右よりの葉(leaf：枝を持たない末端の節）と位置を交換したうえで，ヒープから切り離す。これは，単純選択法において，探し出した最大値を末尾の要素と交換したうえで，次の検索範囲を狭める操作に相当する。

3）　葉にあったデータを根に移動したことでヒープの秩序は乱れてしまっている。そこで，根に置いたデータを適切な位置までふるい落とす。具体的には，左右の子のうち，より大きなほうと比較してみて，子のほうが大きければ，互いの位置を交換する（子を1段持ち上げて，自分は空いた場所に落ちる），という操作を繰り返し，適切な格納場所を探す。

4）　2)～3)で1パス。以下，同様の操作を木が空になるまで繰り返す。

このアルゴリズムでは，最初にヒープを構成することができさえすれば，各パスでは最大でも$2\log_2 n$回（＝木の高さ－1の2倍）の比較と$\log_2 n$回（＝木の高さ－1)の交換によって，最大値を選び出すことができる。各パスの計算量は$\log_2 n$のオーダーであり，それをn回繰り返すわけだから，ヒープソートの計算量は$n\log_2 n$のオーダーとなる。

このアルゴリズムを実現するにあたって，まず，データ構造をどうするか考えよう。一般に2分木は，データそのものと左右の子を指すポインタを備えた自己参照的・再帰的な構造体を使って構成する。C風に表すと，

```
struct tree {
    int key;
    struct tree *left;
    struct tree *right;
};
```

のような構造体で1データを表し，leftで左側の子，rightで右側の子をポイントする形になる。だが，ヒープはうまい具合に配列で疑似的に表現することができる。具体的には，木の根に配列の先頭要素を，根の左側の子に第2要素，右側の子に第3要素，というように，配列の各要素を上→下，左→右の順序で対応づけていく。図3にその様子を示す(便宜上，図3では配列の添え字を1から数えている)。見てのとおり，ある節と配列の1要素a［i］が対応しているとき，その左側の子はa［i×2］，右側の子はa［i×2＋1］に，それぞれ対応する，という規則性がある。

残る問題はどうやって最初にヒープを構成するかだが，巧妙な方法が考え出されている。

図2　ヒープの一例

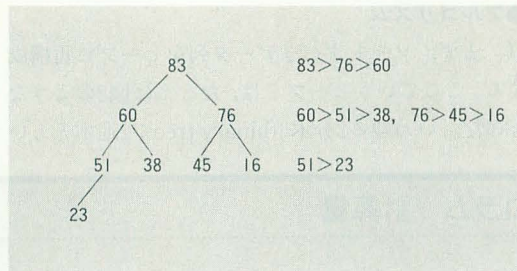



図3　配列とヒープの対応

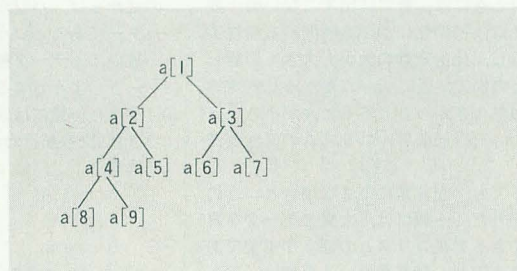
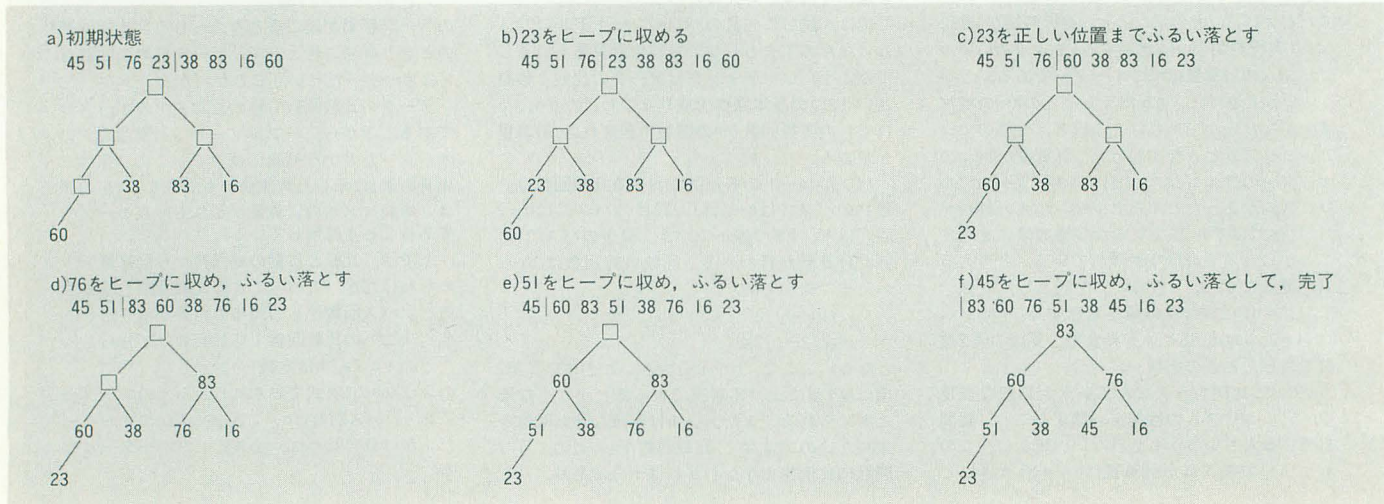



図4　4ヒープの構成

1) 配列を前半部分と後半部分の2つに分ける。配列の後半部分はちょうどヒープの葉にあたるから，とりあえず，これらが正しい位置にあるものと仮定する。また，葉ではない各節は空であると考える。

2) 空の節のうち，なるべく下でなるべく右のものにデータを入れる。配列レベルでいうと，前半部末尾のデータが後半部に入るよう，境界を移動することになる。

3) いま空の節に入れたデータは正しい位置に収まってはいない可能性がある。そこで，先ほど出てきたような方法でふるい落とす。

4) 以下同様に，空の節にデータを入れてはふるい落とす。空の節がなくなった時点で，配列全体はきちんとしたヒープの体裁となる。

図4にヒープを構成する様子を，また，図5にヒープ構成後の，ヒープソート本体の動きを示す。

●プログラム例

配列を疑似2分木として使うときには，ポインタでではなく，添え字を使ってアクセスするのがわかりやすい。しかし，シェルソートのときと同様，添え字からアドレスを計算する手間はなるべく省きたい。そこで，リスト4では，アドレスレジスタa0がつねに配列の先頭アドレスを指すよう固定しておき，例によって4倍（配列の1要素の大きさ倍）した添え字をデータレジスタに入れ，

```
    move.l    0(a0, d4.l), d0
```

のようなインデックスつきアドレスレジスタ間接形式で配列にアクセスする方法をとった。こうしておくと，

```
    add.l     d4, d4              ＊インデックスを2倍
    move.l    0(a0, d4.l), d0     ＊左の子
    move.l    4(a0, d4.l), d1     ＊右の子
```

のように，インデックスを2倍（添え字を2倍することにあたる）することによって，左右の子を参照することができる。もっとも，実際のリストでは，都合によりインデックスがつねに4だけ大きい（添え字を1から数えることに相当する）ので，

```
    move.l    -4(a0, d4.l), d0    ＊左の枝
    move.l    0(a0, d4.l), d1     ＊右の枝
```

のように，その分をディスプレースメントで補正するようにしてある。

リスト4の29行まででヒープ構成の下ごしらえをする。ここでは，d4に“ヒープと未整理部分の境界を指すインデックス”，a2に境界のアドレスそのものを入れている。表現は違っても両者は同じ要素を指すことになる（処理の過程では，その場その場で都合のよいほうを使う）。なお，正確にいうと，29行の時点でのd4は予定よりも要素1個分だけ小さい。これは，続くループの先頭32行でa2をプリデクリメントする関係でのつじつま合わせだ。

32行からのループでヒープを構成する。a2をプリデクリメントして境界を移動し，新たなデータをd0に取り出しては，ふるい落としていく。ふるい落としの処理はあとでまた使うので59行以下のサブルーチンsiftにまとめてある。効率を優先した結果，siftへは，

a3＝ふるい落とし開始位置のアドレス
d0＝ふるい落とすデータ
d3＝(a3)の左の子を指すインデックス

をレジスタで渡すようになっている。

図5 ヒープソート

a)初期状態
83 60 76 51 38 45 16 23

b)根の83を末端の23と交換し，23をふるい落とす
76 60 45 51 38 23 16|83

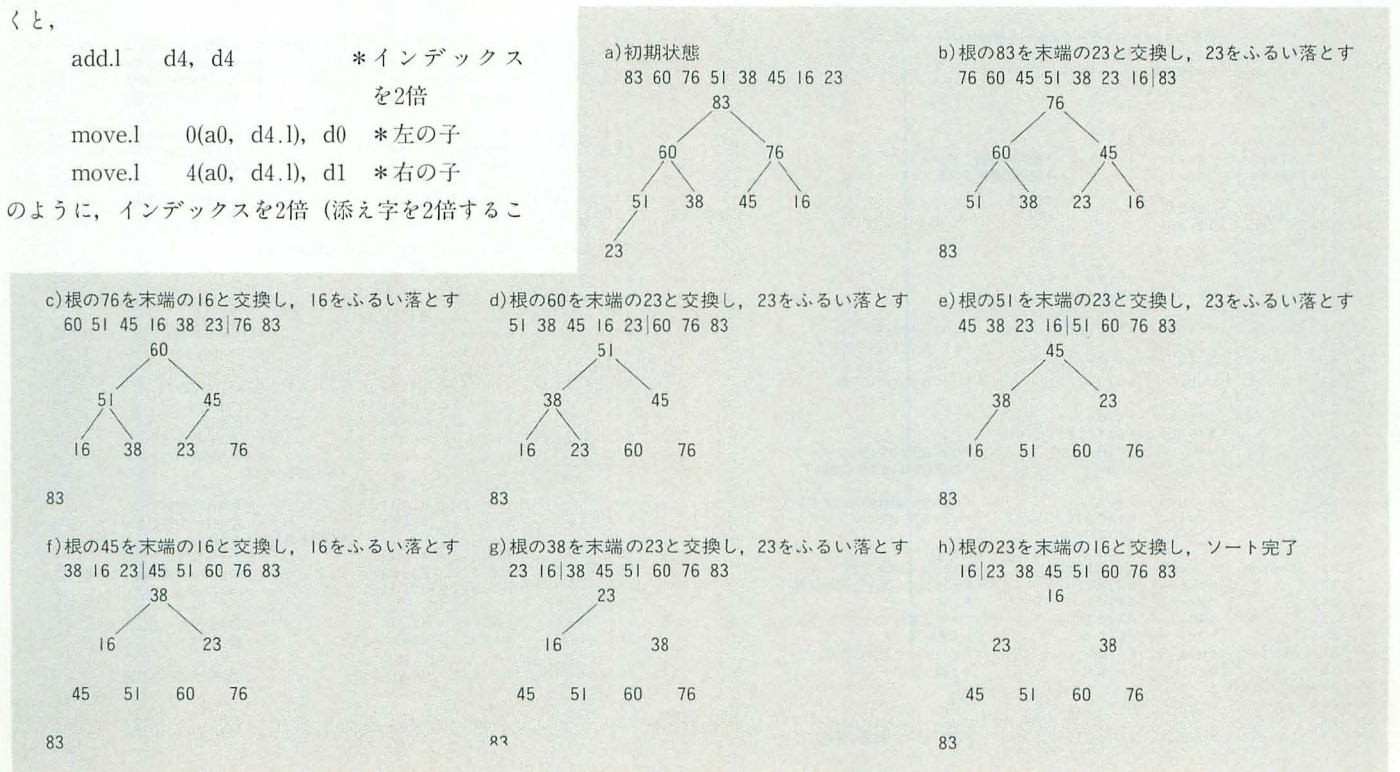
c)根の76を末端の16と交換し，16をふるい落とす
60 51 45 16 38 23|76 83

d)根の60を末端の23と交換し，23をふるい落とす
51 38 45 16 23|60 76 83

e)根の51を末端の23と交換し，23をふるい落とす
45 38 23 16|51 60 76 83

f)根の45を末端の16と交換し，16をふるい落とす
38 16 23|45 51 60 76 83

g)根の38を末端の23と交換し，23をふるい落とす
23 16|38 45 51 60 76 83

h)根の23を末端の16と交換し，ソート完了
16|23 38 45 51 60 76 83

4) このようにxを決めると，ソート済みの配列をソートするときに最高速が出る。

siftの中身では，さきほどの説明のとおり，子との大小関係を比較しつつ，データが収まるべき位置を探している。ここでは，注目している節が持つ子の数に応じ，

1) 子が0個（葉）
2) 子が1個（右の子が存在しない節）
3) 子が2個（ほとんどの節）

の3通りに応じた場合分けが必要な点に注意したい。

ヒープが構成できたら，42行以下のループで実際にソートする。アルゴリズムどおり，黙々とヒープの先頭要素（ヒープ中の最大値）と末尾の要素を交換してはふるい落とすという処理を繰り返している。

クイックソート

クイックソート（quick sort）は，まじめに比較とデータ移動を繰り返すアルゴリズムとしては，現時点での最高速を誇るソーティングアルゴリズムだ。平均的な処理時間が$n \log_2 n$に比例するという点ではヒープソート同様だが，余計な初期化の手間がいらず，細部の処理も簡潔な分，ヒープソートよりもさらに2倍は速い（プログラムの作り方にもよるが）。

●アルゴリズム

1) 配列中の適当な要素を“境界値”として選び，仮にxとおく。配列中のデータとまったく無関係にxを選んでもアルゴリズムは正常に働くのだが，配列中に実在する値をxとして使ったほうがあとの処理が簡単になる。なお，配列の“位置上の中央の要素”をxとして使うことが多いので，ここでもこれに従うことにする[4]。

2) xより小さいデータを配列の左（先頭方向）に，大きいデータを配列の右（末尾方向）に移動することで配列を2つの区間に分割する。ここまでが1パス。

3) 各区間に要素が1個しか含まれていない状態になるまで，分割したそれぞれの区間に対して1)，2)を繰り返し，さらに分割していく。

3) 各区間に要素が1個しか含まれていない状態になったとき，ソートは完了している。

配列を2分する処理は，次の手順で行える。

1) 配列の両端を指すポインタを用意する。便宜上，左側のポインタをL，右側のポインタをRとする。

2) 左側から順にポインタLをインクリメントしつつ配列を走査して“x以上”のデータを探す。ここで，xとして配列中に実在する値を使ったメリットが現れる。xは走査区間中のどこかに存在するわけだから，ポインタLが配列の右端を越えてしまう前にかならず見つかる。ということは，わざわざLが右端を越えてしまったかどうかのチェックをする必要がない（xは番人として働く）。

3) 右側からは，ポインタRをデクリメントしつつ，“x以下”のデータを探す。

4) この時点で，L<Rが満たされていなければ(ポインタがすれ違ってしまっていたら)，分割は終了している。LとRがすれ違ったところが，分割の境目と

リスト4 SORT9.S

```
 1: *            符号つき32ビットデータの配列を昇順にソートする
 2: *            (ヒープソート)
 3: *            8000要素のソートに 1.37秒
 4: *
 5:          .xdef   sort
 6: *
 7:          .offset 8
 8: ATOP:    .ds.l   1          *配列の先頭アドレス
 9: ABOT:    .ds.l   1          *配列の最終アドレス+1
10: *
11:          .text
12:          .even
13: *
14: sort:
15:          link    a6,#0
16:          movem.l d0-d4/a0-a4,-(sp)
17:
18:          movem.l ATOP(a6),a0-a1   *a0=配列先頭アドレス
19:                                   *a1=配列末尾+1アドレス
20:          move.l  a1,d4            *
21:          sub.l   a0,d4            *d4=配列のバイト数
22:
23:          lsr.l   #1,d4            *
24:          moveq.l #$fffffffc,d0    *
25:          and.l   d0,d4            *d4=配列中央のインデックス
26:          beq     done             *要素数が1以下なら即終了
27:
28:          subq.l  #4,a1            *a1=配列最終要素のアドレス
29:          lea.l   0(a0,d4.l),a2    *a2=配列中央のアドレス
30:
31:                           *ヒープを形成する
32: loop0:   move.l  -(a2),d0         *d0=ふるい落とすデータ
33:          movea.l a2,a3            *a3=ふるい落とし開始位置
34:          move.l  d4,d3            *
35:          add.l   d3,d3            *d3=左の子のインデックス
36:          bsr     sift             *ふるい落とす
37: next0:   subq.l  #4,d4            *インデックス>0のあいだ
38:          bgt     loop0            *  繰り返す
39:
40:          bra     next1
41:                           *ソートする
42: loop1:   move.l  (a1),d0          *d0=ヒープ末端の要素
43:                                   *  =今度ふるい落とすデータ
44:          move.l  (a0),(a1)        *ヒープ中の最大要素を
45:                                   *  ヒープ末端に移動し
46:          subq.l  #4,a1            *  ヒープから切り離す
47:
48:          movea.l a0,a3            *a3=ふるい落とし開始位置
49:          moveq.l #2*4,d3          *d3=左の子のインデックス
50:          bsr     sift             *ふるい落とす
51:
52: next1:   cmpa.l  a1,a0            *ヒープが空になるまで
53:          bcs     loop1            *  繰り返す
54:
55: done:    movem.l (sp)+,d0-d4/a0-a4
56:          unlk    a6
57:          rts
58: *
59: snext:   move.l  d1,(a3)          *子を1段引き上げる
60:          movea.l a4,a3            *自分は1段落ちる
61:          add.l   d3,d3            *d3=次の子のインデックス
62:
63: sift:    lea.l   -4(a0,d3.l),a4   *(a4)=左の子
64:          move.l  (a4),d1          *d1=左の子
65:          cmpa.l  a4,a1            *子の数は?
66:          bcs     sdone            *  子はいなかった
67:          beq     sskip            *  左だけだった
68:                           *左右の子がいる
69:          move.l  4(a4),d2         *d2=右の子
70:          cmp.l   d2,d1            *どちらが大きい?
71:          bge     sskip            *  左の子の方が大きかった
72:                           *右の子の方が大きい
73:          addq.l  #4,d3            *右の子を指すように
74:          addq.l  #4,a4            *  インデックスとポインタを修正
75:          move.l  d2,d1            *d1=右の子
76:
77: sskip:   cmp.l   d1,d0            *子の方が大きいあいだ
78:          blt     snext            *  繰り返す
79:
80: sdone:   move.l  d0,(a3)          *ここが収まるべき位置
81:          rts
82:
83:          .end
```

なる。

5)　さもなくば，Lの指すデータとRの指すデータを交換し，2)以降を繰り返す。

この手順では，最終的に，

　　左側にx以下の値
　　右側にx以上の値

がくるよう分割される。見てのとおり，xそのものはどちらにも含まれる可能性がある。そのため，ときには無駄な交換が発生することもある[5]。本当は，

　　左側にx未満の値
　　中央にxと等しい値
　　右側にxより大きい値

と3分割するか，せめて，

　　左側にx未満の値
　　右側にx以上の値

または，

　　左側にx以下の値
　　右側にxより大きい値

と分割するようにしたほうが効率はよい。ただ，そうすると，xが番人として働かなくなるから[6]，上の2)や3)のループの終了条件を複雑にしなければならず，かえって処理速度を落とすことになるだろう。

クイックソートは，おおむね$n\log_2 n$に比例する時間で仕事をする。しかし，最悪のケースでは，処理時間はn^2に比例するまでに落ちる。選んだ境界値xが区間中の最小値であった場合を考えよう（そうなる確率は1/n)。その場合，区間は，最小値1つからなる左の区間と，残りn－1個の要素からなる右の区間に分割される。その右区間を分割するときにも，また，最小値を境界値として選んでしまったとする(その確率は1/｛n－1｝)。この不運が無数に重なると，n回の分割操作が必要となり，1パスごとに最小値を順に選ぶ“バブルソートの世界”に落ち込んでしまう。滅多[2]に起こらないことではあるが，クイックソートを利用するうえでは，この“最悪の事態”を頭に入れておく必要がある。

では，図6に境界値xとして区間中の中央の要素を使う場合の分割の様子(1パス分)，図7に最終パスまでのソートの様子を示す。

●プログラム例

クイックソートはアルゴリズムがそもそも再帰的であり，プログラムも再帰呼び出しを利用するとすっきり書ける。つまり，分割した区間をさらに分割するために，クイックソートルーチン本体をサブルーチンコールするわけだ。では，さっそくプログラム例を見てもらおう（リスト5)。

まず，25行までで，中央の位置を指すインデックス（＝ソート範囲のバイト数の半分）をd0に求める。このとき，区間に要素数が1個しか残っていない場合をはじいている（再帰の終了条件)。

それから，28行で，境界値xとして使う配列の中央にあるデータをd0に取り出し，29～30行でポインタL（a1）とR（a2）の初期化をする。Lは注目している要素を直接指すが，Rのほうはプリデクリメントする都合で，注目する要素の直後を指すようになっている。

36行から分割処理のループが始まる。37～38行で左側から，39～40行で右側から，位置の狂っている要素を探し出す。見つけた時点で，ポインタLとRがすれ違っていたらループを抜ける(42～43行)。まだすれ違っていないようなら，2つの要素を交換し（33～35行)，処理を繰り返す。

分割処理のループを抜け，45行に達したときにはポインタL（a1レジスタ）は分割の境界を要素1個分過ぎたところを指しているから（ポストインクリメントしているため)，その分を補正する。この段階で配列は，

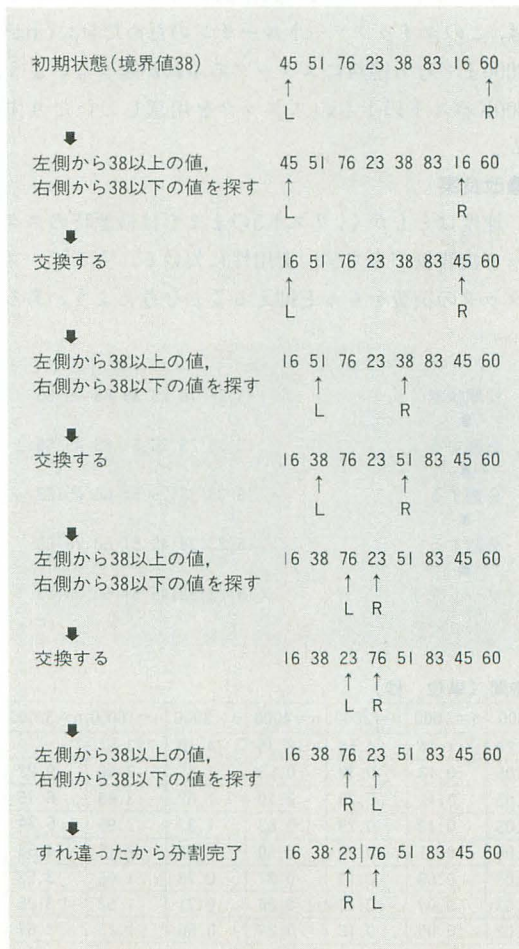
　　a0以降a1の直前
　　a1以降a3の直前

の2つの区間に分割された。以下，47～49行で左側の区間，51～53行で右側の区間をそれぞれさらに分割するために，クイックソートルーチン自身を再帰的

5)　LもRもxと等しい値をポイントしているときにも交換が起こる。

6)　配列の全データが同じ値の場合を考えてみるとよい。

図6
クイックソートにおける分割の様子

```
初期状態(境界値38)              45 51 76 23 38 83 16 60
                                ↑                    ↑
                                L                    R
  ↓
左側から38以上の値,              45 51 76 23 38 83 16 60
右側から38以下の値を探す         ↑                 ↑
                                L                 R
  ↓
交換する                        16 51 76 23 38 83 45 60
                                ↑                 ↑
                                L                 R
  ↓
左側から38以上の値,              16 51 76 23 38 83 45 60
右側から38以下の値を探す            ↑        ↑
                                   L        R
  ↓
交換する                        16 38 76 23 51 83 45 60
                                   ↑        ↑
                                   L        R
  ↓
左側から38以上の値,              16 38 76 23 51 83 45 60
右側から38以下の値を探す               ↑  ↑
                                      L  R
  ↓
交換する                        16 38 23 76 51 83 45 60
                                      ↑  ↑
                                      L  R
  ↓
左側から38以上の値,              16 38 76 23 51 83 45 60
右側から38以下の値を探す               ↑  ↑
                                      R  L
  ↓
すれ違ったから分割完了           16 38 23|76 51 83 45 60
                                      ↑  ↑
                                      R  L
```

に呼び出す。

さて，再帰呼び出しは大量のスタック消費をともなう。リスト5のクイックソートルーチンを利用するときには，あらかじめ，メインルーチン側で十分な大きさのスタックを確保しておかなければならない。どれだけ必要なのか，計算してみよう。

リスト5ではサブルーチン呼び出し1回ごとに，ロングワードデータ10個分（40バイト）のスタックを消費する。内訳は以下のとおり。

引数に2
サブルーチンからの戻りアドレスに1
link時のa6の待避に1
レジスタ待避に6

これに再帰呼び出しの最大ネスティングレベル（入れ子の回数）を掛ければ，必要なスタック量が求まる。

配列を分割する過程で，毎回ちょうど綺麗に2等分できれば，ネスティングレベルは$\log_2 n+1$となる。要素数が2^{20}個あったとしても，21段しかネストしない。しかし，分割が非常に偏り，毎回，要素1個の区間と残り全部に分割されるような“最悪の事態”では，ネスティングレベルはnに等しい。これを見越すと，結局，n×40バイトのスタックを用意する必要がある。先月のリスト1の動作試験用のプログラムでは，このクイックソートルーチンのためだけに，nが8000までなら絶対にスタックあふれが起きないよう，300Kバイト以上ものスタックを用意していたりする。

●改良案

速度はともかく，リスト5のままでは最悪時のスタック消費量が多すぎ，実用性に欠ける。そこで，スタックの消費をもっと抑えることを考えよう。ある程度までは，コーディング上の工夫で逃げることができる。リスト6に，なるべくレジスタの待避を行わないようにしたバージョンを示そう。リスト6のスタック消費量は最悪の場合でもリスト5の40％に抑えられている。

サブルーチンはsort，qsortの2つに分割した。sortはメインルーチンからただ1度呼ばれ，レジスタを保存し，処理を下位のqsortに引き継ぐというだけの働きをする。qsortは自分自身を再帰的に呼び出すクイックソートルーチン本体で，こちらでは，引数を積む以外にはスタックを使わない。link命令すら削ってしまった。qsort内で，レジスタをプッシュしているのは58行の，

```
movem.l   a0/a1/a3,-(sp)
```

だけだ。ここでは，62行のサブルーチンコールに備えてa1とa3を待避しつつ，直後の59行のサブルーチンコールのために，引数a0とa1をスタックに積んでいる。a1のプッシュに2つの意味があるところがミソといえばミソだ。

もっと抜本的な改良を加えたのがリスト7。これは非再帰版のクイックソートルーチンで，再帰呼び出しをループに展開してある。分割した各区間のうちの片方はその場で処理し，もう一方は区間の両端アドレスをスタックに積んでおいて，あとでスタックから取り出して処理する。スタックが空になった時点でソートは完了している。再帰的なサブルーチン呼び出しをなくしたことで，スタック消費が抑えられるし，実行速度も上がる。

これに加えて，リスト7ではほんのわずかな速度低下と引き換えに，スタック消費量を大幅に抑える魔法を使っている。55〜64行のあたりでやっているように，分割した区間のうち，長いほうを選んでスタックに積み，短いほうをその場で処理するようにすれば，スタックには最大$\log_2 n$個分の区間しか積まれずにすむのだ[7]。nが2^{20}であったとしても，20段×8バイト＋α（サブルーチン冒頭でのレジスタ待避の分など）で足りる計算だ。これでようやく，このクイックソートルーチンも実用レベルに達したといえるだろう。気をよくして，改めて高速化を図る。

一般に高級なソートアルゴリズムは，要素数nが小さいときには威力を発揮できず，処理が複雑な分，かえって単純な方法よりも遅くなるものだ。もちろん，クイックソートも例外ではない。要素が十分少ないときには，クイックソートではなく，もっと単純な方法（たとえば，単純挿入法）を使おう，という常識的な結論が導き出される。

ここで，クイックソートの再帰的な性格を考えよう。ソートする配列の要素数が非常に多くても，処理が進むにつれ，配列はごく短い多数の区間に分割

7） リスト7では，左右の区間の長さを求めて比較するのではなく，分割の境界が区間の中央のどちら側にあるかによって，左右の区間の長短を調べている点に注目。

図7　クイックソート

初期状態	45 51 76 23 38 83 16 60
↓	
分割する	16 38 23|76 51 83 45 60
↓	
分割する	16 23|38|76 51 60 45|83
↓	
分割する	16|23|38|45 51|60 76|83
↓	
ソート完了	16|23|38|45|51|60|76|83

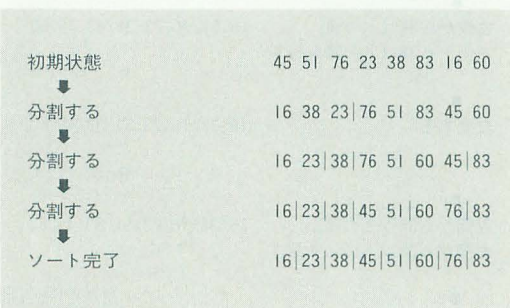



表1　ランダムデータのソート時間（単位：秒）

		n=500	n=1000	n=2000	n=4000	n=8000	n=16000	n=32000
単純挿入法	（リスト1）	0.28	1.16	4.58	18.35	74.10	293.51	---------
シェルソート	（リスト2）	0.05	0.13	0.30	0.69	1.65	3.90	9.22
	（リスト3）	0.05	0.12	0.29	0.70	1.62	3.68	6.75
ヒープソート	（リスト4）	0.05	0.13	0.29	0.63	1.37	2.95	6.34
クイックソート	（リスト5）	0.05	0.11	0.24	0.50	1.04	2.18	4.58
	（リスト6）	0.03	0.08	0.17	0.37	0.78	1.66	3.53
	（リスト7）	0.03	0.07	0.16	0.34	0.71	1.52	3.25
	（リスト8）	0.02	0.05	0.12	0.27	0.56	1.23	2.67

される。クイックソートアルゴリズムは，ソート完了間際になって，苦手な“短い区間の群れ”に襲われ足留めをくらう形になるわけだ。そこで，分割が進み，各区間の要素数が十分少なくなったら，より単純なソートアルゴリズムに切り替えるという高速化案が浮かんでくる。

このアイデアをプログラムにするとリスト8のようになった。リスト8では，区間中の要素数が7行のラベルMで定義された数未満になった時点で，単純挿入法に切り替えている。このプログラムの場合，だいたいM＝15あたりが最適な値のようだ。

プログラムについては，ほとんど見るべき点はない。骨格はリスト7のままで，それに先月のリスト7に若干手を入れた単純挿入法部分を埋め込んだ形をしている。例によって，単純挿入法の部分は番人を置くことでまだ速くなるのだが，たいした速度向上が望めるわけでもなく，メインルーチン側の負担が大きくなるデメリットのほうが大きいと判断し，番人の導入は避けた。その点さえ除けば，リスト8は今日の僕が作りうる，最良のソートルーチンだ(追記：と，思ったらいらない行があった。40行を削除のこと)。

＊

最後に，今回作ったソートルーチンの実行時間を表1に示しておく。先月は，さまざまな傾向を持たせたデータについても時間を計測したが，今回はランダムなデータに限った。各アルゴリズムがどのような傾向のデータを好むか（また，苦手とするか）という点については，読者への研究課題としよう。

ということろで，ソートの話は終わる。次回はグラフィックに舞い戻り，たぶん，ライン描画をやる，と思う。

リスト5 SORT10.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (クイックソート)
 3: *           8000要素のソートに 1.04秒
 4: *
 5:             .xdef   sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l   1           *配列の先頭アドレス
 9: ABOT:       .ds.l   1           *配列の最終アドレス+1
10: *
11:             .text
12:             .even
13: *
14: sort:
15: qsort:
16:             link    a6,#0
17:             movem.l d0-d1/a0-a3,-(sp)
18:
19:             movem.l ATOP(a6),a0/a3      *a0=配列先頭アドレス
20:                                         *a3=配列末尾+1アドレス
21:             move.l  a3,d0               *
22:             sub.l   a0,d0               *d0=配列バイト数
23:             lsr.l   #1,d0               *d0=配列バイト数/2
24:             moveq.l #$ffff_fffc,d1      *下位2ビットをマスクする
25:             and.l   d1,d0               *d0=中央のインデックス
26:             beq     done                *要素数が1以下なら即終了
27:
28:             move.l  0(a0,d0.l),d0       *d0=境界値
29:             movea.l a0,a1               *a1=左端ポインタ
30:             movea.l a3,a2               *a2=右端ポインタ
31:             bra     next1
32:                             *分割する
33: loop1:      move.l  -(a1),d1            *2要素を交換する
34:             move.l  (a2),(a1)+          *
35:             move.l  d1,(a2)             *
36: next1:
37: loop2:      cmp.l   (a1)+,d0            *左側から位置の狂った
38:             bgt     loop2               *  要素を探す
39: loop3:      cmp.l   -(a2),d0            *右側から位置の狂った
40:             blt     loop3               *  要素を探す
41:
42:             cmpa.l  a1,a2               *ポインタがすれ違うまで
43:             bcc     loop1               *  繰り返す
44:
45:             subq.l  #4,a1               *行きすぎた分を補正する
46:                             *分割完了
47:             movem.l a0/a1,-(sp)         *左側を再帰的に分割する
48:             bsr     qsort               *
49:             addq.l  #8,sp               *
50:
51:             movem.l a1/a3,-(sp)         *右側を再帰的に分割する
52:             bsr     qsort               *
53:             addq.l  #8,sp               *
54:
55: done:       movem.l (sp)+,d0-d1/a0-a3
56:             unlk    a6
57:             rts
58:
59:             .end
```

リスト6 SORT11.S

```
 1: *           符号つき32ビットデータの配列を昇順にソートする
 2: *           (クイックソート：スタックをなるべく使わない版)
 3: *           8000要素のソートに 0.78秒
 4: *
 5:             .xdef   sort
 6: *
 7:             .offset 8
 8: ATOP:       .ds.l   1           *配列の先頭アドレス
 9: ABOT:       .ds.l   1           *配列の最終アドレス+1
10: *
11:             .text
12:             .even
13: *
14: sort:
15:             link    a6,#0
16:             movem.l d0-d2/a0-a3,-(sp)
17:
18:             movem.l ATOP(a6),a0/a3      *a0=配列先頭アドレス
19:                                         *a3=配列末尾+1アドレス
20:             moveq.l #$ffff_fffc,d2      *d2=下位2ビットマスクデータ
21:
22:             movem.l a0/a3,-(sp)         *引数を積み直し
23:             bsr     qsort               *  クイックソート本体を呼ぶ
24:             addq.l  #8,sp               *
25:
26:             movem.l (sp)+,d0-d2/a0-a3
27:             unlk    a6
28:             rts
29: *
30: qsort:
31:             movem.l 4(sp),a0/a3         *a0=区間先頭アドレス
32:                                         *a3=区間末尾+1アドレス
33:             move.l  a3,d0               *
34:             sub.l   a0,d0               *d0=区間バイト数
35:             lsr.l   #1,d0               *d0=区間バイト数/2
36:             and.l   d2,d0               *d0=中央のインデックス
37:             beq     done                *要素数が1以下なら即終了
38:
39:             move.l  0(a0,d0.l),d0       *d0=境界値
40:             movea.l a0,a1               *a1=左端ポインタ
41:             movea.l a3,a2               *a2=右端ポインタ
42:             bra     next1
43:                             *分割する
44: loop1:      move.l  -(a1),d1            *2要素を交換する
45:             move.l  (a2),(a1)+          *
46:             move.l  d1,(a2)             *
47: next1:
48: loop2:      cmp.l   (a1)+,d0            *左側から位置の狂った
49:             bgt     loop2               *  要素を探す
50: loop3:      cmp.l   -(a2),d0            *右側から位置の狂った
51:             blt     loop3               *  要素を探す
52:
53:             cmpa.l  a1,a2               *ポインタがすれ違うまで
54:             bcc     loop1               *  繰り返す
55:
56:             subq.l  #4,a1               *行きすぎた分を補正する
```

```
57:                              *分割完了
58:         movem.l a0/a1/a3,-(sp)  *左側を再帰的に分割する
59:         bsr     qsort           *
60:         addq.l  #4,sp           *
61:
62:         bsr     qsort           *右側を再帰的に分割する
63:         addq.l  #8,sp           *
64:
65: done:   rts
66:
67:         .end
```

リスト7　SORT12.S

```
 1: *       符号つき32ビットデータの配列を昇順にソートする
 2: *       (クイックソート:非再帰版)
 3: *       8000要素のソートに 0.71秒
 4: *
 5:         .xdef   sort
 6: *
 7:         .offset 8
 8: ATOP:   .ds.l   1               *配列の先頭アドレス
 9: ABOT:   .ds.l   1               *配列の最終アドレス+1
10: *
11:         .text
12:         .even
13: *
14: sort:
15: qsort:
16:         link    a6,#0
17:         movem.l d0-d2/a0-a4,-(sp)
18:
19:         movem.l ATOP(a6),a0/a3  *a0=配列先頭アドレス
20:                                 *a3=配列末尾+1アドレス
21:         moveq.l #$ffff_fffc,d2  *d2=下位2ビットマスクデータ
22:         clr.l   -(sp)           *スタックの底マーク
23:         bra     loop0
24:
25: rerty:  movea.l d0,a0           *a0=区間先頭
26:         movea.l (sp)+,a3        *a3=区間末尾+1
27:
28: loop0:  move.l  a3,d0           *
29:         sub.l   a0,d0           *d0=区間バイト数
30:         lsr.l   #1,d0           *d0=区間バイト数/2
31:         and.l   d2,d0           *d0=中央のインデックス
32:         beq     next0           *要素数が1以下なら分割完了
33:
34:         lea.l   0(a0,d0.l),a4   *a4=区間中央
35:         move.l  (a4),d0         *d0=境界値
36:         movea.l a0,a1           *a1=左端ポインタ
37:         movea.l a3,a2           *a2=右端ポインタ
38:
39:         bra     next1
40:                              *分割する
41: loop1:  move.l  -(a1),d1        *2要素を交換する
42:         move.l  (a2),(a1)+      *
43:         move.l  d1,(a2)         *
44: next1:
45: loop2:  cmp.l   (a1)+,d0        *左側から位置の狂った
46:         bgt     loop2           *  要素を探す
47: loop3:  cmp.l   -(a2),d0        *右側から位置の狂った
48:         blt     loop3           *  要素を探す
49:
50:         cmpa.l  a1,a2           *ポインタがすれ違うまで
51:         bcc     loop1           *  繰り返す
52:
53:         subq.l  #4,a1           *行きすぎた分を補正する
54:                              *分割完了
55:         cmpa.l  a4,a1           *分割された区間のうち
56:         bcc     right           *  短いほうを先に処理する
57:
58: left:   movem.l a1/a3,-(sp)     *右区間をスタックに積んでおき
59:         movea.l a1,a3           *左区間の分割を
60:         bra     loop0           *  先に行う
61:
62: right:  movem.l a0/a1,-(sp)     *左区間をスタックに積んでおき
63:         movea.l a1,a0           *右区間の分割を
64:         bra     loop0           *  先に行う
65:
66: next0:  move.l  (sp)+,d0
67:         bne     rerty
68:
69: done:   movem.l (sp)+,d0-d2/a0-a4
70:         unlk    a6
71:         rts
72:
73:         .end
```

リスト8　SORT13.S

```
 1: *       符号つき32ビットデータの配列を昇順にソートする
 2: *       (クイックソート:非再帰・単純挿入法併用版)
 3: *       8000要素のソートに 0.56秒
 4: *
 5:         .xdef   sort
 6: *
 7: M       equ     15              *区間中の要素数がM未満になったら
 8:                                 *単純挿入法に切り替える
 9: *
10:         .offset 8
11: ATOP:   .ds.l   1               *配列の先頭アドレス
12: ABOT:   .ds.l   1               *配列の最終アドレス+1
13: *
14:         .text
15:         .even
16: *
17: sort:
18: qsort:
19:         link    a6,#0
20:         movem.l d0-d3/a0-a4,-(sp)
21:
22:         movem.l ATOP(a6),a0/a3  *a0=配列先頭アドレス
23:                                 *a3=配列末尾+1アドレス
24:         moveq.l #$ffff_fffc,d2  *d2=下位2ビットマスクデータ
25:         moveq.l #M*4,d3         *d3=切り替え点*4
26:         clr.l   -(sp)           *スタックの底マーク
27:
28:         bra     loop0
29:
30: rerty:  movea.l d0,a0           *a0=区間先頭
31:         movea.l (sp)+,a3        *a3=区間末尾+1
32:
33: loop0:  move.l  a3,d0           *
34:         sub.l   a0,d0           *d0=区間バイト数
35:         cmp.l   d3,d0           *残り区間が十分短くなったら
36:         bcs     isort           *  単純挿入法に切り替える
37:
38:         lsr.l   #1,d0           *d0=区間バイト数/2
39:         and.l   d2,d0           *d0=中央のインデックス
40:         beq     next0           *要素数が1以下なら分割完了
41:
42:         lea.l   0(a0,d0.l),a4   *a4=区間中央
43:         move.l  (a4),d0         *d0=境界値
44:         movea.l a0,a1           *a1=左端ポインタ
45:         movea.l a3,a2           *a2=右端ポインタ
46:
47:         bra     next1
48:                              *分割する
49: loop1:  move.l  -(a1),d1        *2要素を交換する
50:         move.l  (a2),(a1)+      *
51:         move.l  d1,(a2)         *
52: next1:
53: loop2:  cmp.l   (a1)+,d0        *左側から位置の狂った
54:         bgt     loop2           *  要素を探す
55: loop3:  cmp.l   -(a2),d0        *右側から位置の狂った
56:         blt     loop3           *  要素を探す
57:
58:         cmpa.l  a1,a2           *ポインタがすれ違うまで
59:         bcc     loop1           *  繰り返す
60:
61:         subq.l  #4,a1           *行きすぎた分を補正する
62:                              *分割完了
63:         cmpa.l  a4,a1
64:         bcc     right
65:
66: left:   movem.l a1/a3,-(sp)     *右区間をスタックに積んでおき
67:         movea.l a1,a3           *左区間の分割を
68:         bra     loop0           *  先に行う
69:
70: right:  movem.l a0/a1,-(sp)     *左区間をスタックに積んでおき
71:         movea.l a1,a0           *右区間の分割を
72:         bra     loop0           *  先に行う
73: *
74: isort:                          *単純挿入法
75:         lea.l   -4(a3),a2       *
76:         bra     inext           *
77: iloop:  move.l  (a1)+,d1        *
78:         cmp.l   d1,d0           *
79:         ble     found           *
80:         move.l  d1,-8(a1)       *
81:         cmpa.l  a3,a1           *
82:         bcs     iloop           *
83:         addq.l  #4,a1           *
84: found:  move.l  d0,-8(a1)       *
85: inext:  movea.l a2,a1           *
86:         move.l  -(a2),d0        *
87:         cmpa.l  a0,a2           *
88:         bcc     iloop           *
89: *
90: next0:  move.l  (sp)+,d0
91:         bne     rerty
92:
93: done:   movem.l (sp)+,d0-d3/a0-a4
94:         unlk    a6
95:         rts
96:
97:         .end
```

シミュレーションプログラミング入門 第4回

冬だから，シミュレーション

今回はお湯の温度制御を題材に連続型シミュレーションのお話をします。微分とか積分とかの数式がボコボコと出てきますので，心して読んでください。まあ，Oh!Xの読者の皆さんならそんなことは心配無用かな？

Kamon Masato 華門 真人

冬にはやっぱり雪が似合う。雪がどんどん降るので，ひとりで悦に入っている。もっともこれだけ降ると，吹雪の中であわや凍え死に，なんて目にもあう。

凍死してしまった場合はどうするか。これはもう温泉しかないでしょう。いまやアフタースキーといえば「温泉」の時代。それも露天風呂にかぎる。

体は熱いぐらいなのに，頭だけがひんやりと冷たい。この感覚がたまらない。日頃はバーボン党の僕もこのときだけは熱燗である。ただ，ここでひとつ注意しなければならないことがある。

温泉といってもこんなときは硫黄泉は避けるべきである。さもないとあとで「あなた，なんか臭いわよ」といわれ，涙を飲むことになりかねない。

万が一，硫黄泉に遭遇してしまった場合は上がる際に体を真水でよく洗っておくことをお勧めする。

とまあ，冬といえばスキーに温泉，華やかなものである。ゲレンデは色とりどりのウェアでいっぱい。そして温泉はといえば，うら若い美女との混浴（ということを期待しているとたいていそこにいらっしゃるのは50年ぐらい前の美女である，しょせん現実は厳しい）。

冬における天（国）がスキーと温泉だとすると地（獄）はさしずめテニスであろう。春から夏にかけて眩しいばかりの若さに満ち溢れていたはずのテニスコートは，いまや人影もまばらである。

無理もない。冬のテニスは単に寒いだけである。誰かさんの素敵なおみ足も何重にも覆われて，かすかに浮かぶ足のラインにその名残を留めるだけだ。

寒くて辛い冬のテニス，誰が好き好んでやるものだろうか。アフターテニスで温泉というのも期待薄だし……（逆にいえば夏のスキーヤーも間抜けである。特に，冬は王様のように暮らしているインストラクター，通称「イントラ」はまさしく陸に上がった河童と化す。季節労働者とは彼らのための言葉である。もっとも最近は海外スキーという大技もあるので，あまり馬鹿にはできない）。

しかし，冬に努力したものこそがうまくなれるというのもまた事実である。マリンスポーツでも同じだ。ついでにつけ加えると受験も同じだろう。辛いときに頑張ったものこそが最終的に勝利を勝ちとれるのだ（といいつつ，必ずしもそうでもないのが受験の面白いところである）。

control

先日，寒い中を久し振りにテニスの練習に出た。僕のラケットはガットではなく，蜘蛛の巣を張っているといえばどれだけ久し振りかがわかるだろう。

元来，僕はテニスも好きな人間である。当然合宿などにもよく参加する。軽井沢に毎年2回は行く民宿があるのだが，ここがまたなかなか居心地のいい宿である。

宿のご主人とも顔馴染みだから多少騒いでも怒られることはないし，テニスをやって騒いでという生活にはもってこい。

ところが，そこはだいぶ古い民宿，施設という面ではずいぶん見劣りがする。タコ部屋はまだ許せるとして，どうも困るのがお風呂である。

どうやら給湯設備に根本的な問題があるらしい。なかなかお湯が出ないのである。というわけで，風呂に入るとまず蛇口の奪い合いになる。確保したいのは給湯パイプに近いほうの蛇口である。

これならば必要なだけはお湯が出る。しかし，この蛇口を確保しそこねると事態は深刻である。給湯パイプから遠のくにつれて等比級数的にお湯が出なくなってくる。

少しでも同じ量がコンスタントに出てくるのならば，まだましである。気長に待てばいいだけのこと。が，現実はそうはいかない。

現実はといえば，お湯の量が劇的なまでに変化するのである。具体的にいうと，自分のところより給湯パイプに近い蛇口が閉じられたとたん，その分のお湯が自分の蛇口に集中する。それに気がつかずにそのお湯を浴びると，ああ恐ろしい，熱湯を浴びる羽目になる。

これはまさしく制御が追いついていないいい例である。しかし，ところ変わって都内の某高級ホテルのスイートルームともなると，一度温度をセットするだけで快適なお湯を好きなだけ楽しむことができる。こちらはうまく制御されたいい例といえるだろう。

Heater again

前回はその高度な温度制御を連続型シミュレーションによってシミュレートしてみた。今回はいかにシミュレーションを実現するかということを中心にして見ていこうと思う。

連続型のシミュレーションというのは，数式によって状態が記述される点に特徴があった。逆にいえばその数式を解くことによって連続型のシミュレーションを実現できることになる。

それではまず，シミュレーションの対象である温水器について復習し，数式に表してみることにしよう。やっぱり，初めは「お

絵描き」から。モデルを絵で表すと図1のようになる。これを見ればひと口にモデルとはいっても大きく2つに分けられることがわかるだろう。

すなわちコンピュータによる**制御**そのものと，制御されている温水器すなわち**制御対象**の2つだ。まずは後者，制御対象から考えてみることにしよう。

前回見てきたように，このモデルの制御対象には2つの特性がある。覚えていらっしゃるだろうか，そう，1次遅れとムダ時間のことだ。

先に簡単なムダ時間から片づけることにしよう。ムダ時間というのは入力されたモノが出力されるまでにいくばくかの時間を要するということから生ずるタイムラグのことだ。

これを数式で表すとどうなるか。これは簡単だ。ムダ時間をτとすると，出力yはムダ時間τの分だけ前の入力xに等しい。ということは，

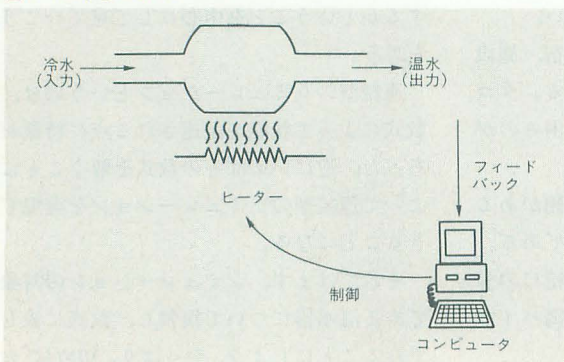
数式1

$$y(t)=x(t-\tau)$$

となる。これでモデルを構成する数式がまずひとつできたことになる。これを数式1としよう。

さて，次は制御対象のもうひとつの特性，1次遅れについて考えてみる。実はこの1次遅れというのが今回のヤマ場である。覚悟はよろしいかな。まあ何はともあれ，1次遅れというモノがどのようなモノだったかを思い出さねばならない。

1次遅れというのは入力がダイレクトに出力に反映しない特性のことである。入力が徐々に出力に出るような特性ともいえる。前回のやかんの例をもう一度思い返してみてほしい。

この特性を数式で表現しなければならない。結果からいうと入力をx(t)，出力をy(t)とすると，

数式2

$$\frac{dy(t)}{dt}=k(a\cdot x(t)-y(t))$$

という一見複雑そうな微分方程式で表される。

ここでk,aは方程式の係数を表している。先月のプログラムのパラメータk,aはここから来ている。もう完全に拒絶反応を起こして失神している方もいらっしゃるかもしれない。が，安心してほしい，この微分方程式を正面から解くつもりはないのだから。

とりあえず，上の式（数式2としよう）の意味なるものを考えてみようか。左辺は出力y(t)を微分したものだ。いくら数学が苦手な人でも，微分が速度を表すことぐらいは覚えていらっしゃるだろう。

なに，覚えていないって。それではいい機会だから復習しておくように。ともかく，左辺は出力y(t)の変化速度を表すものであることがわかった。では，右辺はどうなのだろう。

実は右辺のほうが簡単で，入力x(t)と出力y(t)の差に係数を掛けたものだ。ということは式全体ではどんな意味があるのだろうか。左辺は出力の変化速度，右辺は入力と出力の差を表している。結局，出力の速度は入力と出力の差（正確にいえば，入力に係数の掛かったものと出力の差）に比例するということになる。

といってもわかりにくいかな。つまりこういうことだ。入力と出力の差が大きければ，出力は速く変化する。逆に小さければ，出力はゆっくりと変化する。

ある一定の入力を与えたときの1次遅れの出力について考えてみれば，よりわかりやすいだろう（前回の図3を参照）。最初は出力が小さい，ということは入力と出力の差が大きいので出力は速く変化する。つまり，出力は入力に追いつこうとして激しく変化する。

しかし，そうやって出力が大きくなってくると，今度は逆に入力と出力の差が小さくなり，出力はゆっくりとしか変化しなくなる。つまり，入力に少しずつしか近づこうとしなくなってしまう。

図1 温水器モデル

これで数式2の意味はわかった。では，どうやって解くか。まあ，そう焦ることはない。解法はのちほどみっちりと学ぶことになるのだから。

ここはとりあえず，制御のほうについてもモデルを構成している数式を明らかにしてみようと思う。

formula

前回，制御にはP,I,Dという3種類の制御方式があることは述べた。今回はまず，それぞれを数式に表現してみよう。

トップバッターはP制御，すなわち比例制御である。比例制御とは何か。ひと言でいえば「現在の値と目標値の差（偏差）に応じた」制御ということになる。それでは数式では？　これは思ったより簡単だ。偏差をd(t)，制御量をz(t)とすると，制御量は偏差に比例しているのだから，

数式3

$$z(t)=kp\cdot d(t)$$

とすればよいだけのことだ（これを数式3とする)。ここでkpというのは比例制御の係数を表している。

さて，トップバッターをあっさりと打ち取って，迎えたのは2番バッターI制御(積分制御）である。選手名鑑によるとI制御は「偏差に応じた速度での」制御を得意技とするらしい。かなりの技巧派のようだ。

とはいえ，制御の速度が偏差に比例しているということだから，

$$\frac{dz(t)}{dt}=ki\cdot d(t)$$

でOKだ。言うまでもなく，左辺が制御の速度(微分は速度を表しているんだよね)，右辺が偏差である。当然，kiというのは積分制御の係数ということになる。

もちろん，このままの形の式でもいいのだが，どうも見にくい気がする。というのも左辺が微分の形になっているため，肝心の制御「量」がどうなっているのかがわかりにくい。

さすが技巧派，やっぱりこのままではダメだ。第一，数式3との形の釣り合いもとれない。ということで，少し変形を試みてみようと思う。どうするか。といっても左辺を“z(t)＝”の形にしたいのだから，微分

を普通の形に戻すためにはどうすればよいのかを考えてみればいい。

微分の反対は，そう積分だ。ということは，微分されたものを積分してやれば普通の形に戻してやることができる。それでは左辺だけ積分する，というわけにもいかない。そんなことをしたら両辺のバランスが崩れてしまう。

左辺だけがダメなら両辺とも積分してやればよい。そう，これが正解。ということで両辺ともに積分すると，

数式 4

$$z(t)=ki\cdot\int d(t)\,dt$$

となる。これでめでたく積分制御の式になる(数式4)。見ればわかるように，今度は右辺が積分の形になっている。実はこの積分が，「積分制御」という名のいわれなのだ。

さて9回裏，なんとか2番バッターもフェンスぎりぎりのレフトフライに打ち取った。いよいよ3番バッターD制御の登場である。D制御の得意技はといえば，「偏差の変化速度に応じた」制御であるらしい。どうやらこいつもテクニシャンのようだ。

しかし，I制御を始末したいまとなってはもう恐いものはない。制御が偏差の変化速度に比例するのだから，速度は微分というセオリーさえ思い出せば，苦もなく，

数式 5

$$z(t)=kd\cdot\frac{dd(t)}{dt}$$

という式を導き出すことができる。

この式では，右辺が偏差の微分の形になっている。これが，「微分制御」と呼ばれるゆえんである。

ゲームセット。これで数式探しの旅は終わった……。

technique

しかし，まだ先は長い。今度はその数式をコンピュータ上で実現するという作業が必要になってくる。ここで，いままでに出てきた数式を表にまとめてみた。表1を見てほしい。

この温水器モデルを実現するためには，表1にあるモデルを構成する5本の数式をコンピュータ上に実現しなければならない。数式をコンピュータ上で実現する場合，素直に記述してうまくいく場合と，うまくいかない場合がある。うまくいくよい例が数式1だろう。

数式1におけるムダ時間τだけ前の入力$x(t-\tau)$をx_0，出力$y(t)$をyとしてやれば，常に，

$$y=x_0$$

という代入文でこと足りてしまう。

ところが，数式2ではそうはいかない。式の中に微分が含まれているからだ。ご存じのように，我々の使っているコンピュータは通常「デジタル」である（ごく一部にアナログコンピュータというものも存在してはいるが)。

そしてデジタルコンピュータではまともに微分を取り扱うことはできない。それではどうすればよいのだろうか。まず考えつくのが，微分方程式を解いて普通の式の形に戻す，という方法だ。

たとえば，

$$\frac{dy(t)}{dt}=2x$$

というような微分方程式であれば，

$$y(t)=x^2+C$$

と簡単に解が見出せるから，この解の式をプログラムすればことはすむ。

しかし，このようなやり方は必ずしもうまくいくわけではない。まともな方法では解くことのできない微分方程式も多く存在するし（というより簡単に解ける微分方程式というのは実は貴重な存在である)，なによりも微分方程式を解くためには専門的かつ高度な知識を必要とされる。

そこで「技」が必要とされる。どんな場合でも，解ける方法。しかも，簡単でなければならない。その技こそが，数値積分と呼ばれるものである。この数値積分の長所は，デジタルコンピュータとも非常に相性がよいという点である。

では，ここで一般的な数値積分について説明することにしよう。微分が速度だとすると，積分は何を表すのだろうか。そう，面積を表しているのだ。図2を見てほしい。関数$x(t)$がある。この関数$x(t)$を積分した$\int x(t)\,dt$はいったい何を表しているのだろうか。

これは図2の斜線部，すなわち$x(t)$の面積を表すことになる。これに注目して考えてみると，積分値を求めるということは，面積を求めるということに置き換えて考えることができるのがわかるだろう。

一般に積分をそのまま解くことは難しいが，面積を求めることは比較的容易である。そのいちばん簡単な方法が，いわゆる「台形法」と呼ばれるものである。

図2のいちばん右のほう（逆向きの斜線になっている部分）に注目していただきたい。この部分の面積は，上底が$x(t)$，下底が$x(t-\omega)$，高さがωの台形の面積にほぼ等しい。

ここで，“ほぼ等しい”というのが曲者である。ほぼ等しい，すなわち，これは近似である。もうおわかりかもしれないが，数値積分というのはまともには解くことが難しい式を近似によって解こうという方法だ。

近似とはいえ，精度はまずまずである。というのも台形の高さωを小さくすることによっていくらでも精度を上げることができるからである（ωを小さくすればそれだけ誤差が減る)。

さてこうして，一部分でも面積が求まればしめたものである。それらを全部足してやれば求めたい全面積，つまり積分値が得られたことになる。

さらにコンピュータ流の式で表してみる

表1　モデルを構成する数式

数式1（ムダ時間）
$y(t)=x(t-\tau)$

数式2（I次遅れ）
$\frac{dy(t)}{dt}=k(a\cdot x(t)-y(t))$

数式3（比例制御）
$z(t)=kp\cdot d(t)$

数式4（積分制御）
$z(t)=ki\cdot\int d(t)\,dt$

数式5（微分制御）
$z(t)=kd\cdot\frac{dd(t)}{dt}$

図2　数値積分

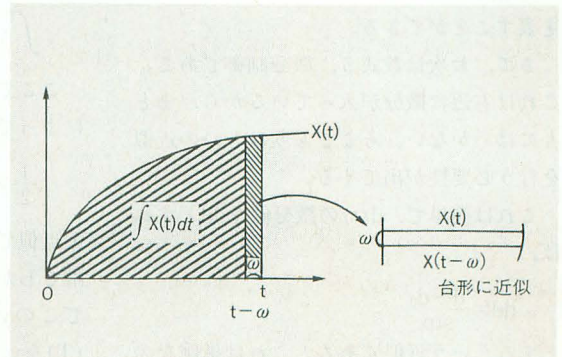
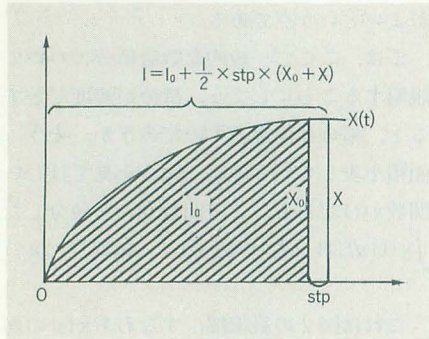



図3　図2を具体化

ことにしよう。図3のように，ある地点tまでの積分値を求めたい場合，数値積分する単位であるステップをstp，stp時間だけ前のxの値をx_0とすれば，積分値Iは，

$$I=I_0+\frac{1}{2}\times stp\times(x_0+x)$$

と表すことができる。ただしここで，I_0はstp時間前までの積分値である。

さて，これでどんな式でもどんとこい，といえるようになった。とはいえ，数式2はちょっとばかり複雑なのであと回しにすることにしよう（真打ちはあとからやってくる）。

ということは，次は数式3になる。しかしこの式は微分も積分も含んでいないからそのまま素直に，

$$z=kp\times d$$

と表すことができる。

ふがいない，と思いつつ次は数式4である。ここで先ほどの数値積分が活用される。右辺の積分$\int d(t)\,dt$は数値積分に従えば，

$$id=id_0+\frac{1}{2}\times stp\times(d_0+d)$$

となる。ただし，ここでid_0はstp時間前までの積分値，doはstp時間前の偏差d(t)の値を表している。

こうして得られた積分idを用いれば，積分制御は，

$$z=ki\times id$$

と表すことができる。

さて，お次は数式5，微分制御である。これは右辺に微分が入っているから，まともにはいかない。そこでもうひとつの近似を行う必要性が出てくる。

これは簡単で，d(t)の微分値をddとすれば，

$$dd=\frac{d-d_0}{stp}$$

とするという近似である。これは単純なのであまり細かくは述べないが，

微分＝変化速度＝傾き

ということを思い出せば，すぐに理解できるであろう。

この近似による微分値ddを用いれば，微分制御は，

$$z=kd\times dd$$

となる。

さあ，こうして制御をコンピュータ上の式として表すことが可能になった。これまでの結果を利用すれば，PID制御の制御量は，

$$z=kp\times d+ki\times id+kd\times dd$$

と表せることになる。

construction

機は熟した。いまや残っているのは数式2のみである。これを数値積分を用いて攻略し，コンピュータ上の式として表せれば，いよいよプログラムを作ることができる。

さて，その数式2であるが，このままでは解くことができない。そこで両辺を積分し，積分の形にもっていく。そうすると晴れて数値積分を用いることができる。

では，まず数式2の両辺を積分してみよう。得られた式は，

$$y(T)-y(0)=k\times(a\int_0^T x\,dt-\int_0^T y\,dt)$$

となる。こうすると左辺は普通の形になり，右辺に2つの積分が生ずる。この右辺の積分を数値積分を用いて解いてやればよいということになる。

この2つのうち，xの積分は数値積分で簡単に求まる。これをixとしておこう。問題はyの積分である。y自体を求めたいのだから，通常はyの積分など求められるはずがない。しかたがないので，とりあえず分解してみよう。

$$\int_0^T y\,dt=\int_0^{T-stp} y\,dt+\int_{T-stp}^T y\,dt$$

ここで，右辺第2項は数値積分（台形法）によってさらに，

$$\frac{1}{2}\times stp\times(y(T-stp)+y(T))$$

と近似できるが，これらのうち時刻Tの段階でわからない値はy(T)のみである。そこでこのy(T)を左辺にまとめてしまう。y(T)をy，y(T−stp)をy_0とすると，y(0)が0であれば，

$$y(1+\frac{stp}{2}\cdot k)$$
$$=k(a\cdot ix-(iy_0+\frac{stp}{2}\cdot y_0))$$

となる。かくして，ようやくyが求まるというわけだ。

いよいよすべての数式の解釈が終わった。あとはこれをモデルに即してプログラムすればいい。

プログラムを作り上げるためには，まず，図1のモデル図を元にフローチャートを作らねばならない。この際にだいたいの変数名も決めてしまおう。こうしてできたフローチャートが図4である。

さらにこのフローチャートに，数式をコンピュータ流に割りつける。

ヒータ出力値y2はフィードバックされた制御zによる。ただし，y2とzとの間には1次遅れの関係が成立している。ヒータ出力値y2がやがて出力値y3になるわけだが，ここでムダ時間が生ずる。今回はムダ時間τを，プログラムの簡単化のために時間単位stpと等しい値とした。

そして，y3と目標値targetとの差が偏差d（deviation）になり，この偏差dに応じて制御量zが変化する。

この関係に従って変数の間に数式が割り当てられる。そしてようやくプログラムの骨組みが出来上がるのである。

それでは実際のプログラムを見てみよう。リスト1はX-BASIC用のもの。基本的に先月のX1用と同じである。3000行台まではいわゆるおまじない，初期設定である。

ここで変数名の説明をしておこう。できるだけ本文に近い形の変数名にしたが，そこは僕のやること，結構いい加減である。とりあえず目安になるものを書いておこう。

・パラメータはいいだろう。先月説明したとおり

・頭にiがついていれば，積分値であることを示す

・頭に*d*がついていれば，微分値であることを示す

・最後に$_0$がついていれば，stp（＝ムダ時間）だけ前の値であることを示す

・dl（delay）とあれば，1次遅れのかかった値であることを示す

まあ，だいたいこんなものだろう。あくまで目安のつもりで参考にしてほしい。

さて重要な部分は3500行台だ。ここで得られた数式を元に現在の状況を算出している。上に出てきた数式が，実際にどのように記述されているのか，調べてみるとよいだろう。

So what？

こうしてひととおり連続型シミュレーションの作成方法を見てきたわけだが，もう一度全体を振り返ってみることにしよう。

こうして見てきて，仕事の手順が離散型とかなり似ていることに気がついただろう。まず，始めにモデルの概念図がある。これをしっかりと把握したあとでモデルのフローチャートを作ることになる。

そして，そのフローチャートに実際のプログラムを割りつけていけばよいわけだが，ここで大きな差が出る。離散型ではトランザクションの移動を中心に考えた。連続型ではまずモデルを構成する（連続した）数式を考え，その数式をプログラムへと落としていくことになる。

ただ，数式といってもそのままではコンピュータ上で実現できないことも多い。特に積分や微分が問題となる。そこでうまく近似を使ってやることにより，コンピュータで処理できる形にしてやる，という手間も必要になってくる。

もっとも近似を使わないで処理する方法もあることはある。ラプラス変換やz変換などを使えば微分方程式をうまく処理できる(こともある)。しかし，これらは当然かなりの知識を必要とする。もし，使えるならばラプラス変換などを試してみるのもいいだろう。

こうしてなんやかんやの手間を経て，ようやく連続型のシミュレーションが完成する。So what，それで？

シミュレーションには目的がある。これは離散型でも連続型でも同じことだ。シミュレーションが完成したらいろいろと実行条件を変えてどのような差が出てくるかを観察してみたい。

これだけ自分の思ったように操作できるのはシミュレーションの特権だ。むちゃくちゃな条件でも，シミュレーションはちゃんとした答えを出してくれる。現実ではこうはいかない。

変なパラメータによって，熱湯が吹き出してしまったらどうなることか。誰も現実の火傷の責任はとってくれない。

話がそれたが，パラメータを操作することによって，P制御，PI制御，PID制御を実際に確かめることができる（パラメータは前回を参照してほしい。本当はよい制御を行うためのパラメータの最適化なども考えねばならないのだが，今回は本筋から外れるので省略した。詳しく知りたい人は先月の参考文献などを参照するといいだろう）。制御が高度化するにつれて，早く目的値に達し，しかも安定することがわかるだろう。

なかでもPID制御は優れた制御性能を示す。当たり前といえば当たり前の話。PID制御は化学プラントなどでも実際に活用されているのだ。

ところが，前回でも少し触れたが，このPID制御をうわ回る制御が存在するというのだ。いまをときめく（もうちょっと古びてきたが）ファジィ制御がそれである。

fuzzy

PID制御をうわ回る性能とは，具体的にどういうことなのだろうか。温水器の例でいえば，より早く目標値に達し，しかも安定的であるということになる。本当にそんなことが可能なのだろうか。

可能かどうかを知るためには，まずファジィ制御とはどんなものかを知らねばならないだろう。

もうご存じだと思うが，ファジィとは「曖昧」ということである。すなわちファジィ制御とは曖昧な制御ということになる。といっても，まだそれこそ曖昧である。それでは，実際にファジィ制御の核となっているルールの例を見てみることにしよう。

ファジィ制御では次のようなルールに従って制御を行っている。

「もし速度が速く車間距離が狭ければ，減速せよ」。

ここでファジィがファジィたるところは，「速ければ」とか「減速せよ」という曖昧な言葉が使われているところにある。このルールにおいては，速さの基準や，減速の基準がはっきりと示されていない。

これが普通の言語（普通のBASICなど）であれば，上のルールは，

　　if　速度＞80キロ
　　　and/or　車間距離＜20メートル
　　　　then　10キロ減速

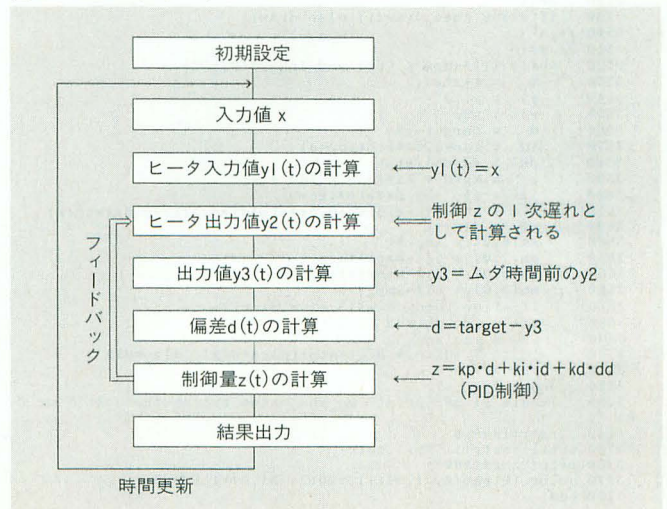
のように表される。すなわち，速さや減速の程度がはっきり示される。

一方ファジィでは，ただ速いとはいっても60キロでも速いと見なされる場合があるし，逆に90キロでも速くはないとなることもある。これはほかのルールなどにも応じて「曖昧に」決まってくるのである。同じ速度でも車間距離に応じて速さの基準というものが変わってくる。

たとえば同じ時速100キロでも，車間距離が200メートルあれば速いとはいえない。逆に車間距離が10メートルしかなければ，これは危険なほど速いということになる。このようにファジィでは基準というものが曖昧になっている。言い換えればそれだけ人間の感覚に近いということになる。

それではこの差は制御においてはどのように出てくるだろうか。ファジィでない場合，境界線がはっきりしているため，振動を起こしやすく，あまり安定しない。

つまりこういうことだ。たとえばさっきの減速ルールの場合，通常の制御であれば，80キロをちょっとでも上回ったら，すぐ

図4　フローチャート

にガクンと10キロも減速してしまう。以降これの繰り返しになり，70キロと80キロの間を振動する。

ところがファジィの場合，ちょっと速い場合はちょっと減速するという曖昧さを持ち合わせている。当然，非常に安定しやすくなるのである。

この安定しやすさは，言い換えれば高度な制御が可能であるということになる。より人間に近いとはよくいわれていることであるが，正確にいえばより柔軟な制御である，ということができるだろう。

さて，だいたいファジィの概要はつかめたことだろう。しかし，本当にファジィを理解するにはやはり実際にファジィ制御にトライしてみるのがいちばんの近道だ。

とはいえファジィ制御を実際にやるのは大変なことだ。いくら家に冷蔵庫が2台あっても，ひとつを無理やりファジィ制御に改造してしまうのは(面白そうだけれども)あまり感心しない。それに皆が体験できなければ意味がない。

NEXT

そこでシミュレーションが登場するのである。実際に作るのが難しいのならば，シミュレートしてやればよいのだ。

今回はコンピュータの中に架空のPID制御の温水器を作り上げた。同じようにファジィ制御だってシミュレートできるはずだ。

都合のいいことに，ある投稿があった。なんと「ファジィ制御による連続車速制御シミュレータ」というまさしくこの連載にぴったりの内容である。

というわけで，次回はこの投稿を全面的にフィーチャーしてファジィ制御のシミュレーションに挑戦してみるつもりである。

ところで，この投稿にかぎらずありとあらゆる投稿，ご意見を歓迎する。ファジィ制御のシミュレーションの次はニューロコンピュータのシミュレーションなんていうのが狙い目かもしれない。

まあ，まだ冬である。寒い現実を忘れ，暖かい部屋でコンピュータの中に(暖かい)現実を作るのもいいんじゃないだろうか。

というわけで来月は北国の温泉から，ファジィ制御についてお送りする。

リスト1

```
1000 /* Simulation model C1 ver.1.00
1010 /*
1020 /*  PID Controller for X-BASIC
1030 /*
1040 /*           1991.1 (C) Cammon
1050 /*
2000 /* initialize
2010 width 96
2020 screen 2,0,1,1
2030 console 0,32,0
2040 color [0,rgb(0,31,31)+1,rgb(0,25,0),65535]
2050 int i
2060 char scls=1,sline=1
2070 float target=80
2080 float k=0.1#,a=1
2090 float kp=45,ki=5
2100 float kd=0.5#,kc=0
2110 float stp=0.1#
2120 char ampx=100,ampy=5
2130 float t
2140 float idlzo,ido,izo
2150 float y1,y2,y3
2160 float d,id,dd
2170 float z,iz,dlz,dlzo
2180 float y2o,do,zo
2190 str rstr[96]
2200 dim int  mm(3,1)
2210 dim char mc(3)={ 0,3,11,5 }
3000 /*
3010 repeat
3020   set()
3030   if scls then wipe()
3040   t=0
3050   idlzo=0: ido=0: izo=0
3060   y1=x(t): y2=y1: y2o=y1: y3=y2o
3070   do=target-y3: zo=kp*do+kc
3080   mm(0,1)=0: mm(1,1)=511-ampy*target
3090   mm(2,1)=511-ampy*x(t): mm(3,1)=511-ampy*y3
3100   for i=0 to 3
3110     mm(i,0)=mm(i,1)
3120   next
3130   if sline then drawl() else draw()
3500 /*
3510 /* main
3520   while ((t*ampx < 767) and inkey$(0)="")
3530     t   = t+stp
3540     y1  = x(t)
3550     y3  = y2o
3560     d   = target-y3
3570     id  = ido+0.5#*stp*(do+d)
3580     dd  = (d-do)/stp
3590     z   = kp*d + ki*id + kd*dd + kc
3600     iz  = izo + 0.5#*stp*(zo+z)
3610     dlz = k*(a*iz-(idlzo+0.5#*stp*dlzo))/(1+0.5#*stp*k)
3620     y2  = dlz+y1
3630     mm(0,0) = ampx*t
3640     mm(1,0) = 511-ampy*target
3650     mm(2,0) = 511-ampy*x(t)
3660     mm(3,0) = 511-ampy*y3
3670     if sline then drawl() else draw()
3680     ido = id: do = d
3690     izo = iz: zo = z
3700     idlzo = idlzo + 0.5#*stp*(dlzo+dlz): dlzo=dlz
3710     y2o = y2
3720   endwhile
3730   locate 51,18: print "Do you wanna finish the job (Y or C
R) ";
3740   rstr=inkey$
3750 until instr(1,"Yy",rstr)
3760 print chr$(&HB)
3770 color [0,rgb(0,31,31)+1,rgb(31,31,0)+1,65535]
3780 end
5000 /*
5010 /* drawing result
5020 func draw()
5030   for i=1 to 3
5040     pset(mm(0,0),mm(i,0),mc(i))
5050   next
5060 endfunc
5500 /* drawing result - line
5510 func drawl()
5520   for i=1 to 3
5530     line(mm(0,1),mm(i,1),mm(0,0),mm(i,0),mc(i))
5540     mm(i,1)=mm(i,0)
5550   next
5560   mm(0,1)=mm(0,0)
5570 endfunc
6000 /*
6010 /* set constant
6020 func set()
6030   str tstr
6040   repeat
6050     cls
6060     locate 48,16: print "┌";strin(21,"─");"┐"
6070     for i=1 to 11
6080       locate 48,16+i
6090       print "│";space$(42);"│"
6100     next
6110     locate 48,28: print "└";strin(21,"─");"┘"
6120     scls   = setsub("CLS",51,20,scls)
6130     sline  = setsub("LINE",72,20,sline)
6140     target = setsub("Target",51,21,target)
6150     stp    = setsub("Step",51,22,stp)
6160     kp     = setsub("Kp",51,23,kp)
6170     ki     = setsub("Ki",72,23,ki)
6180     kd     = setsub("Kd",51,24,kd)
6190     kc     = setsub("Kc",72,24,kc)
6200     k      = setsub("k",51,25,k)
6210     a      = setsub("a",72,25,a)
6220     ampx   = setsub("Xamp",51,26,ampx)
6230     ampy   = setsub("Yamp",72,26,ampy)
6240     locate 51,18: print "Are you sure (CR or N) ";
6250     tstr=inkey$
6260   until (tstr=chr$(&HD))
6270 endfunc
6280 /*
6290 func str strin(j;int,tstr;str)
6300   int i
6310   rstr=""
6320   for i=1 to j
6330     rstr=rstr+tstr
6340   next
6350 return(rstr)
6360 endfunc
6500 /*
6510 func float setsub(tstr;str,px;char,py;char,tvar;float)
6520   str ttstr
6530   ttstr=left$(tstr+"     ",6)
6540   color 10
6550   locate px,py: print ttstr;" : ";tvar
6560   color 3
6570   locate 51,18: print ttstr;
6580   input " :  ",tvar
6590   color 6
6600   locate px,py: print ttstr;" : ";space$(11)
6610   locate px+9,py: print tvar
6620   color 3
6630   locate 51,18: print space$(41)
6640   return(tvar)
6650 endfunc
7000 /*
7010 func float x(t;float)
7020   return(20)
7030 endfunc
```

今月もいろいろなジャンルのゲームが出揃いました。皆さんお待ちかねの情報もちゃんと用意。それと今月からCDやビデオの紹介記事もやることになりました。こちらのほうの感想もよろしくね。

メルヘンメイズ

ナムコより発売のアーケードゲーム「メルヘンメイズ」がSPSによって移植された。登場するキャラクターが魅力のクォータービュータイプのシューティングアクションゲーム。最初のうちは，シャボン玉をぽんぽん飛ばしながら敵を弾き飛ばして，軽快に進んでいくだけで楽しめるが，先へ進むと足場は狭くなるし，敵の弾の数は増えてくる。可愛さあまって憎さ百倍（？）といえるほど，かなりハードなゲーム構成なのである。

で，気になる移植の出来はというと，現時点のサンプルではほぼ完璧に仕上がっている。問題があるとすれば，キャラクターが多くなると，若干アーケード版より重く感じられる程度である。発売までにもう少し間があるので，改善されることをSPSに期待しておこう。その他のグラフィック，BGM，ゲーム性ともに文句なしといったところ。そうそう，このゲームも要2Mバイトになりそうなので，1Mバイトの人はすぐに増設しよう。　（純）

話題のソフトウェア

今月の情報。まずは，SPSの**メルヘンメイズ**。ゲーセンで大人気だったナムコのゲームです。これはもうすぐ発売される予定。

ひさしぶりのNCSは**シグナトリー**というアドベンチャーを発表。3月末に発売される予定。詳しくはあとのページを見てね。

お待ちかねの**シムアース**，今年の秋には発売されるもよう。このほかイマジニアでは**パワーモンガー**の発売も決定しました。これはもうちょっとあとの話かな。そうそう，シムシティーのペイントツールともいえる**テレインエディター**も4月に発売されるそう。ファンにはたまらないですね。

工画堂スタジオではパズルアクションゲーム**サブナック**を開発中。4月発売予定です。後ろのページを見てね。じゃ，また。

初チャートは初登場だらけ

		(前回順位)
1	ソルフィース	初
2	パロディウスだ！	3↑
3	ラグーン	1↓
4	シムシティー	5↑
5	銀河英雄伝説II	初
6	エメラルドドラゴン	初
7	生中継68	初
8	ナイアス	2↓
9	ワールドスタジアム	初
10	ダンジョン・マスター	4↓
	イメージファイト	10→

いやーはっは。「'91年最初のランキングに乞うご期待！」といっといてお休みしてしまいましてどうもどうも。先月の'90年年間ランキングのほうはお楽しみいただけましたでしょうか。

では，あらためて今月のチャート発表。間があいただけあって，だいぶ入れ替わっています。では，初登場ソフトのコメントを一挙掲載しますぞ。

まずウルフ・チームのソルフィース。「多関節キャラがよい」「あそこまでうにょうにょ動くと気持ちいい！」「難しいけど面白い」「バランスが取れてるし，演出もいい」「ウルフ・チームのくせによくできている」……おい。

銀河英雄伝説IIは「システムが新しくて遊びやすい」「すぐ終わるのでイライラしなくていい」「ファンにはこたえられない」など。

つぎはエメラルドドラゴン。「絵がきれいで，友人が買えとうるさいから」「なじみやすい」「ストーリーが感動する」「広くて広くて頭がぷちっといきそうなマップ。マップのすみからすみまで歩かしてくれる凶悪なシナリオ」……これって推薦してるのか？

生中継68。「グラフィックがすごい。楽しみだ」「写真を見るだけでも迫力がある」「“のぼ”のトルネード投法を見て」。

最後にワールドスタジアム。「2人プレイで楽しめる。しかし1Mバイトで動かないもんだろうか」「んもー，完璧。データ集もほしいな」「バランスが取れている」。

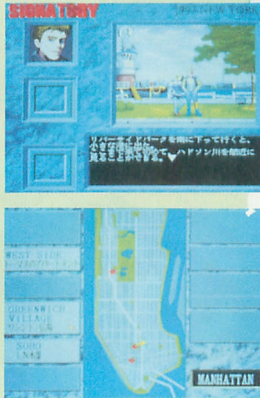
もう誌面がいっぱいになっちゃった。それでは，また来月お会いしましょう。　（浦）

シグナトリー

エルスリードなんかを手掛けていたNCS，最近聞かないなあと思ったら，ひそかに超大作アドベンチャーを制作中だったのだ。それがこのシグナトリー。シグナトリーというのは調印者という意味だ。タイトルからすごそうでしょ。

時は1997年，ニューヨーク。主人公はある組織を追いかけている諜報部員なのだが，捜査を進めていくうちに人類の存在意義に関わる事実に触れていくというものなんだな。

グラフィックもシブめでかっこいいし，画面をクリックしてコマンドを選択する方式もわかりやすい。「MJ-12」とか「第三の選択」という言葉にピクッときちゃう人は今から要チェ〜ックだ！ （浦）

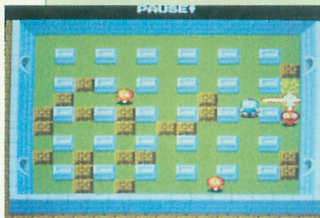
X68000用　5″2HD版5枚組　12,000円(税別)
NCS　☎03(3486)6588

ファランクス

先月の約束どおり，画面写真ができてきたのでお見せしましょう。ザッとこんな感じです。残機＋ダメージ制，ショット4種類×5段階とミサイル3種類，特殊武器4種類だそうです。これだけじゃどのへんがズームらしくなるのかわかんないな。ひょっとしてザコキャラが100発当てないと死ななかったりして。

オープニングデモも届いてます。パネルをいじったりとか，発進のGがかかったとことか，キャラクターの微妙なアニメーションがカッコいい。でも「見せ方」はもうちょっと研究の余地があるかな。

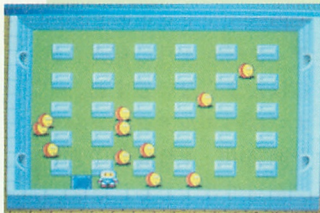
荒けずりだったズームも風格がついてきて，こりゃおじさんゲーム本編が楽しみになってきちゃったな。 （浦）

X68000用　5″2HD版　価格未定
ズーム　☎011(613)0191

ボンバーマン

「ボンバーマン」。なつかしい響きだ。元祖X1の頃の定番ゲームじゃないか。これがなんとシステムソフトからX68000用にバージョンアップして登場するというお知らせ。

敵は人喰い風船。こいつを爆弾で退治するのだ。火力アップのアイテムと爆弾をたくさん置けるアイテムが追加され，敵の種類もグーンと増えたのだ。

面白いのが5人までの対戦モードがついたこと。本当の人間を吹き飛ばすのが危ない快感なのだ。編集部ではさっそくスタッフの間で盛り上がっているから，みんなでワイワイやるタイプのゲームとして，結構いけるかもしれないぞ。（浦）

X68000用　5″2HD版　7,800円(税別)
システムソフト　☎092(752)5278

中華大仙

シャープから発売になったのは，タイトーのコミカルシューティング「中華大仙」だ。

「仙人マイケル・チャンは数々の法術を会得するために修行の旅へ。そこには数多くの試練が待ち受けている。この旅は仙人の最高位である「中華大仙」の称号を得る旅でもある」（パッケージより）。

この仙人，觔斗雲にのって空は飛んじゃうし，弾は打つわ火は吹くわ，暴力的な仙人なのだ。襲い来るのは鳥さんやラーメンドンブリ。どういう修行の旅なんだと言いたくなるようなシュールな世界だな。

ゲームとしては，パワーアップありボスキャラありのツボを押さえた秀作。ゲームセンターでやってた人も，この際おうちに一本いかがです？ （浦）

X68000用　2HD版　7,900円(税別)
シャープ　☎03(3260)1161

マーブルマッドネス

ゲーム史上に残る不朽の名作，マーブルマッドネス。奇妙な世界に棲む奇妙なボールが，ゴールを目指して転がり続ける。待ち受けるトラップ，そして邪魔者たち。頼れるのはおのれのトラックボールさばきのみ。思いどおりに操れそうで操れないボールの操作感覚はまさに至高の快感で，難所を見事クリアしたときの爽快さはなにものにもかえがたい。見たまえこの画面の見事なできばえ。この画面に生命が宿る日，マーブルマッドネスはあらゆるゲーマーの心を魅了することだろう。発売は遅れ気味なので，そのぶんデキに期待しよう。

オンメモリは当然，マウスまたはトラックボールどうしでの対戦までサポートすれば最高。これはささやかな個人的要望。 （A.T.）

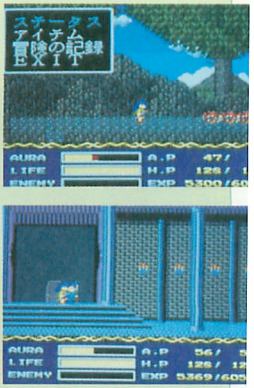
X68000用　5″2HD版　価格未定
ホームデータ　☎078(261)2790

アルガーナ

去年X1ユーザーの間で好評を博したアルガーナ。今度X68000版も登場することになった。

サンプル版を見たところでは，同じM.N.M.の「ティグナスの冒険」に似てるかなって感じ。ただこちらは魔法が使えるので画面はもっとにぎやかだけどね。噂の5重スクロールはもちろんX68000だけあってバッチリ。頭がくらくらしちゃうほど細かく動くのだ。

曲はまだだったんだけど，古代祐三氏のBGMがPSGからFM音源にバージョンアップして聴けるんだから，こりゃ楽しみだよね。 （浦）

X68000用　5″2HD版2枚組　価格未定
ブラザー工業(TAKERU)　☎052(824)2493

Misty vol.7

みんなもう知ってるMistyシリーズ。聞き込みをして自分が推理をするというのが斬新なアドベンチャーゲームだ。新しいシナリオが入ったVOL.7がソフトベンダーTAKERUを通じても発売された。

ディスクの中には，5本のシナリオと神代探偵講座，おたよりコーナーと次号予告編が入っている。さあ，君も名探偵神代龍になって難事件を解決だ。ディスプレイに向かって「犯人は，あなただ！」と指さすのも楽しいぞ。(浦)

X68000用　5″2HD版　5,000円(税別)
X1用　5″2D版　5,000円(税別)
＊TAKERU価格　4,000円(税込)
データウエスト　☎06(968)1236

サブナック

シュヴァルツシルトを出して間もない工画堂から，次作の発表がありました。その名も“サブナック”。画面を見ると，一見RPGかな，と思うのだけれど，ところがぎっちょんちょん，ファンタジーアクションパズルゲームです。

このゲームの目的は，呪いによって石化されてしまった妖精を身に着けている赤いマントの力で呪いを解き，なおかつ神殿に帰してあげること。道中には，これまた石化しているシーフ，戦士，魔法使いがいて，彼らの力を借りてじゃまなモンスターを倒すわけです。がこれはあくまでパズルゲーム，敵を倒すだけでは終わらないのだ。4月発売予定，来月にも紹介するね。(香)

X68000用　5″2HD版2枚組　7,800円(税別)
工画堂スタジオ　☎03(3353)4132

電脳倶楽部

みんな，電脳倶楽部って知ってるかい（笑）。え？　あのえっちなゲーム？　そりゃあんた，妖獣クラブでんがな。

電脳倶楽部はX68000のためのディスクマガジンで，便利なツールや楽しいゲーム，役立つデータ，ビープ音，グラフィックが満載なんだ(宣伝そのまま)。その電脳倶楽部がTAKERUでも買えるようになったぞ。ちょっと値段が高くなっちゃうけど，定期購読する前にちょっと見てみたい人や，現金書留を書くのが面倒臭い人はパソコンショップにダッシュだ！(浦)

X68000用　5″2HD版　1,200円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

日本ファルコム新作試触会

ファルコムが新作を発表！ ロードモナークというタイトルで，あえて分類すればリアルタイムシミュレーションってとこかな。当日の会場は，いろいろな雑誌社さんなどでたいへんな盛況ぶり。とりあえずPC-9801で発売だけどX68000への移植も検討中とのことです。(香)

(善)のゲームミュージックでバビンチョ

えー，今月からOh!Xでもゲームミュージックのでビデオの紹介を（不定期で）することになりました。早速1枚目いってみましょう。

●アルシス・ベストセレクション

ポリスター

前からいっているがアルシスは曲がいい。メチャクチャにいい。Oh!Xスタッフ内にもファンは多い。私などはアルシスの新作が出るたびに，先の曲を聴きたいがために一生懸命ゲームをして，テープに録音していたのだ。他のソフトハウスのゲームミュージックが次々にCD化されていくなか，私はかなりやきもきしていたのだがついに（というかやっと）アルシスのCD「アルシス・ベストセレクション」が発売された。収録内容は「プリンス・オブ・ペルシャ(PC-9801)」，「ナイトアームズ(X68000)」，「スタークルーザー(メガドライブ)」の3作品のBGM。どうやらプリンス・オブ・ペルシャがメインらしく，こちらは未使用曲まで収録されている一方，他2作の曲は全曲収録されていないのが残念。またアレンジバージョンがすこしナニ……（電子楽器のデモプレイみたいだぞ，ありゃ）。でも収録時間は75分と長いし，お買得だな。

・最近アルシスブランドの新作がないけど，どうしたんでしょう。

お勧め度　8

●マージャンサウンドグラフィティ

ポニーキャニオン

「ニチブツの曲はちょっとね……」と視線をそらして冷や汗を流している君！　その気持ちはわかるが，今回発売のは音楽性云々よりも楽しさを追求したアルバムだから安心していいぞ。タイトルからも察しのいくように，ニチブツのマージャンゲームのアルバムなのだ。収録内容はゲーム中のBGMとゲームに出てくる女の子のお喋りだ。

今回のこのCDの目玉は留守番電話用ボイスデータとデートゲーム。前者はゲームに登場している女の子の声で「ただ今私の彼は出掛けています」ってな感じのものが収録されている。きっとこれを実際に留守電に入れて使ったら彼女と大ゲンカになるぞ，うしししし（ホゲー）。で，後者のデートゲームってのは，よくあるゲームブックをCD化したもので「私をドライブに連れていってくれる人はトラック△へ，映画に連れていってくれる人はトラック□へ」といったような内容が延々と収録されているわけ。ハッピーエンドを目指してガンバレ！

・かなり内容は面白いけどカーステレオでボリュームガンガンで聴くのはやめような……。

お勧め度　7

●みつめていいよ2号

ポリスター

セタのスーパーリアルマージャンP3のイメージアルバム。「創刊号」「1号」を含めて今回でシリーズ3枚目。収録内容は同ゲームに登場している女の子のお喋りと歌。今回はテーマが「バレンタイン」ということで大変にぎやかな内容。「私の想い……チョコと一緒に溶かして……」とか「お願い，チョコと一緒に私をもらって……」なんて台詞がポンポンとびかってヘッドホンで聴いてても恥ずかしいぞ，こいつは!! しかしこの調子でLD野球拳ギャルのCDが出たら私は自分の首を引っこ抜いて小脇に抱えてトンボ返りしちまうぞ（編：ホントだな！）。

それにしてもこのゲーム相当人気があるみたいですけど，X68000にどこかのソフトハウスさん，移植したらどうでしょう。4Mバイト要とかいったりしてね。ははは。

・収録時間25分で2,800円はちょっと高いな。

お勧め度　7

終わりに

このコーナーで紹介してほしいゲームミュージックのCDをリクエストしてください。古いものでも結構。買おうと思ってるけど内容が心配とか，ゲーム名はマイナーだけど曲はいいんですとか，凄い大ボケな内容で笑えますとか（初代マリオのCDは凄かったなあ）なにかしらの理由を付けてね。そいじゃ，また。(善)

人類を救えロボキッド！

Yamada Junji
山田 純二

「ジェミニウイング」に引き続き，システムサコムがアーケードゲームを移植。その名は「アトミック・ロボキッド」，家庭用ゲーム機などにも出ているのでおなじみでしょう。なかなか楽しめる仕上がりになっているよ。

システムサコムからのアーケードゲーム移植第2弾「アトミック・ロボキッド」が発売された。原作はUPLという，“手裏剣シュシュシュ”の「忍者くん」を制作したところだ。ジャンルはパワーアップ型シューティング。難易度としてはそれほど難しくないので，肩にサロンパスやパテックスを貼り，目薬片手にプレイする必要はない。割ととっつきやすく，普通の人でも真ん中あたりまで進むことができると思われるレベルである。

そして，移植の出来はどうなっているかというと，これは合格点をあげられる。前回のジェミニウイングと同様，アーケード版に忠実にシステムサコムによる丁寧な移植がされているので，移植を待ち望んでいた人たちは安心していいだろう。

遺伝子は人類を救う ◆◆◆◆◆◆◆◆◆

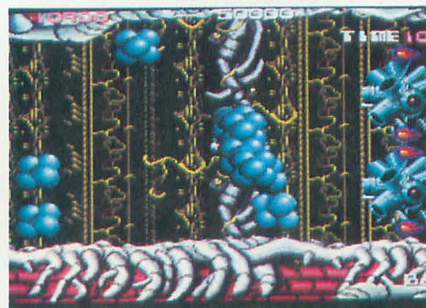
まず，アトミック・ロボキッドのバックグラウンドストーリーを紹介すると，舞台は毎度おなじみの核戦争後の地球である。放射能に汚染された人類を救うため，トミタ博士という人が人類のDNA正常化プログラムを開発。ところが，ロボキッドにその使命を託す前に博士は死んでしまった。はたして，自分の目的もわからずに目覚めてしまったロボキッドが人類を救うことができるのか……。

と，いうのがだいたいのお話。ここで，PCエンジンやメガドライブでこのゲームを遊んだことのある人たちは，「あれ？こんなストーリーだったっけ」，と思ったに違いない。実はこのゲームのアーケード版ではストーリーがなかった，最近では非常に珍しいといえるゲームなのだ。

しかし，移植する側が，ストーリーはないとさみしいとでも思ったのか，勝手にストーリーをつけたためにそういう事態になったようだ。シューティングゲームにストーリーなど無用の長物と思う私としては，「ないものはない」で，すませてしまえばいいと思うけどね。

クリスタルパワー全開 ◆◆◆◆◆◆◆◆

先ほども触れたように，このゲームはパワーアップ型シューティングである。道中に浮いていたり，メタルバードという敵を倒すと出現する“クリスタル”を拾うことにより攻撃パターンが増えていくのである。クリスタルの種類には8種類あり，それぞれ横にどんなものか小さい字で書いてある。クリスタルを攻撃すると，クリスタルの種類がどんどん変わっていく。その性質をうまく使って自分のほしいクリスタルにするのはこのゲームの基本なので，しっかり覚えておこう。

8種類あるクリスタルのうち，武器のクリスタルは4種類ある。いちばん汎用的に使えるのは“3-WAY”で，ボスとの対決のときには貫通力のある“FIRE-2”が結構使える。あと，スティックを入れた方向に攻撃可能なミサイルもある。これは敵の弾も消すことが可能なので狭い通路を進んでいくときなどには非常に便利。最後は広範囲に攻撃可能な“5-WAY”。これは射程距離が短いのが難点。

残りの4種類は武器ではなく，ロボキッドが自由に空を動き回れるようになる“FLY”，自機の速度が倍増する“SPEED UP”，30秒間腕の疲れをいやす“RENSHA!”，そして，“無印”良品のブルークリスタルとなっている。

“FLY”は，ゲームの開始直後の場面に出るもので，とりあえず取らなくてはゲームにならない。ゲームセンターで初めてこのゲームをやったときには，てっきり最初から自由に動き回れるものだと思っていたのでちょっとあせった。これはただのいじわるとしか思えない。

“SPEED UP”は，普段の面では動きが速すぎて必要ないと思うが，ボスとの対決にはあると便利であろう。しかし，一度取ってしまったら死ぬまでその速度でプレイしなくてはならないので，いまいち使えない。使い分けができれば非常に便利だっただろう。

“RENSHA!”は30秒間の連射をしてくれるが，ほとんどの人は連射機能つきのジョイスティックでプレイするであろうから無意味かもしれない。

X68000用　5"2HD版2枚組　8,800円(税別)
システムサコム　☎03(3635)7609

おしくらまんじゅうロボキッド！

せまい通路で弾をかわし，ひたすら撃つ

ブルークリスタルはただ1個だけ取ると1000点のポイントをもらえるだけである。が，塵も積もるとゴミになる，てな具合で4つ集めるとスレッシュホールドなるものが1回使えるようになる。このスレッシュホールドというのは，敵の攻撃を受けてしまったときに一定時間バリアを張ってくれるというもの。スレッシュホールドは3回分までためることができるので，ひととおり武器が揃ってしまったらブルークリスタル集めに励み，あとあとの戦いに備えればいいだろう。

そうそう，このゲームは敵キャラとの当たり判定はない。でも，著しく動きが鈍くなるので無意味な体当たりはしないほうがいい。あと，1回死んでしまうとそのとき使っていた武器がなくなってしまうということも覚えておこう。

挙動不審のロボキッド ◆◆◆◆◆◆◆◆

それにしてもこのロボキッドというキャラクターは，とってもかわいい。2等身のボディにオマケでつけたような足。ワシャワシャと地面を歩き，正面を向いたときにはピョコッと肩をすくめる。“FLY”を取ったあとは空中を自由に動き回れるけど，なんとなく地面をがしがし歩かせて楽しんでしまう。あと，この歩行状態のときに武器の切り替えを行おうとするといきなりジャンプしてしまうので注意しよう。プレイ中，鼻歌ふふふん〜と，気持ちよく遊んでいていきなり敵が出現し，あせって武器の切り替えをしようとしてジャンプしてしまい，敵の弾に突っ込んでしまったということが数回あった。ま，気の緩みがいけないのだが。

第1のボス，FIRE-2で中心部を狙え

にょろにょろ弾を撃つ中ボス，でも弱い

へいロボちゃん！ 何にしやすか？

ステージ構成 ◆◆◆◆◆◆◆◆◆◆◆◆

さて，このゲームはノーマルステージ，中ボスステージ，ボスとの対決，バトルシーンという構成が5サイクル，全部で20面ある。ノーマルステージはひたすら出口を求めて突き進む面。多少入り組んでいる面もあるが，道に迷うことはまずない。ときたま左右とも通路が塞がっているところもある。そんなときには，「下を疑え！」である。

また，ところによって分岐点があり，分岐によって難易度の違う面へ行けるので腕に自信のある人はわざと難しい面に挑戦してみるといいだろう。中ボスステージとは，ひたすら1方向に進んでいき，一応ボスらしきものを倒して進んでいく面であるが，ノーマルステージとあまり変わらない。

ここを過ぎると狭い部屋に閉じ込められてボスとの対決がやってくる。最初は弱いが，ステージが進むにつれ徐々にでかくなり，じわりじわりと触手を伸ばしながら迫ってくるボスを倒すのは少し面倒になってくる。

ボスとの対決にいちばん有効な武器は“FIRE-2”だと思う。ボスにダメージを与えても何の変化もないので多少不安になるが，ちゃんとダメージを与えているはずなので気長に中央部を狙い撃ちしていよう。スレッシュホールドがあるとかなり戦いは楽になる。

ボスを倒したらお次は，敵ロボットとのバトルシーンである。右側にいる敵ロボットを倒せばいいのだが，適当に動きながら撃ってりゃ当たる。ほとんどオマケの面である。

あと，道中には自分の残機数と引き替えに武器をくれる，メカゴジラというのもいる。重なりあってしばらくすると，なぜかショップ画面になるので，自分のほしい武器を選択しよう。ここでは，スレッシュホールドと引き替えにするのがいちばん有効だろう。一見，損をするような取り引きにも思えてしまうが，敵の弾に当たったときに，装備している武器を失わない分だけ得だということはわかると思う。あとは，お決まりの隠れキャラや隠れショップがあるので，各自探してみよう。

さて,なにが楽しい ◆◆◆◆◆◆◆◆◆◆

このゲーム，派手さという点ではバリバリのシューティングゲームに比べると見劣りがする。また，難易度も割と低いので，マニアの人たちにはあまりお勧めできないと思う。見劣りがするといってもデザインセンスのことではなく，節操のないパワーアップシューティングに比べて，という意味なので誤解しないように。

で，僕としてはごく普通の人，シューティングゲームは好きだけどハナモゲに難しいのはイヤ，てな人には勧められる。さすがに後半部分にはきついところがあるが，調子がいいとある程度までさくさく進めるのでストレスがたまらない。それぞれのステージが終わるとあまり意味はないが，デモもあって楽しい。かる〜い気持ちで手軽に遊べるゲームである。

むふふ，次は何だろう

さて，本文中では述べなかった音楽についてだが，きっと忠実に再現されているであろう。オリジナルのほうの曲を聞き込んだことがないのでなんともいえないが，聞いてみたところ普通のゲームミュージックといったところ。例によって，MIDI(MT-32系)にも対応しているので，聞いてみるのもいいだろう。

全体的によくできている。ただひとつの問題は，ボスがやられたときとゲームオーバーのときの赤い弾が飛び散るシーンが下品であるということ。ほかはこれといって問題があるようには思えなかった。

ということで，サコムさん，予告はないがアーケードゲーム移植第3弾はなにをやるのかいまから楽しみにしていますよ。

総合評価	0 … 5 … 10
グラフィック	★★★★★★★★★
普通の人	★★★★★★★★
ロボ＆ゴジラ	★★★★★★★★★★
飛び散る	★★★★★
移植完成度	★★★★★★★★★

一匹狼,宇宙を行く

Kameda Masahiko
亀田 雅彦

スペースフライトシミュレーションとロールプレイングゲームを組み合わせた,「スペース・ローグ」。ポリゴンで処理された3D画面は,ノーマル,チェイス,シネマティックの3種類の視点が選べて,宇宙空間の気分たっぷり。

「ビーッ! ビーッ!」

艦内警報システムが作動,ほとんど同時にベッドから飛び起きた。

「海賊が襲ってきたのか!?」

コクピットのメインスクリーンには旋回中の敵艦が映し出されている。自動航行システムをOFFにして,迎撃体制に入った。

しかし,なにか様子がおかしい。敵はまったく別の方向へ攻撃を仕掛けている。そのとき,緊急回線にタンカーからのSOS信号をキャッチした。どうやら海賊はタンカーを攻撃しているようだ。積荷を奪って売りさばくつもりなのだろう。さかんに降伏勧告を出している。迷った。「いまは大事な任務を遂行中だ。下手な争いに関わりたくはない。しかし,あのタンカーを見捨てるわけにもいかないだろう」。

結局,海賊を攻撃することに意を決した。死闘の末,海賊は倒れた。あなたは帝国が海賊にかけていた賞金を手にした。

オープニング

ご存じの方も多いだろうが,この「スペース・ローグ」は海外からの移植ものだ。制作は,あの「ウルティマ」のオリジン社。他機種には前から発売されていたのだが,今回ようやくX68000にも移植されたしだいである。基本的に「スペースフライトシミュレータ+RPG+アクション」の3つの要素を持ったゲーム。いろいろな星を回って情報を収集したり,自機をパワーアップしたりする。その移動中に敵と遭遇,一転してアクションゲームのノリとなったりする。また,宇宙空間を航行中は画面が3D表示になって,まさにフライトシミュレーターそのもの。お楽しみが盛りだくさんのゲームなのである。

では,オープニングといこう。プレイヤーはプリンセス・ブルーという母艦で航行中,偶然に無人の小型艇(ジョリー・ロジャー号)を発見した。調査のためにひとりで乗り込んでみる。と,そのときレーザー砲の衝撃が船を揺らすのを感じた。母艦がマンチー艦の攻撃を受けているではないか! なすすべもなく母艦は破壊され,仲間は宇宙の塵となってしまった。マンチー艦は去っていく。もはや,残されたものはこのロジャー号しかない。はたしてプレイヤーはどうなるのか!?

というような,絶望的な状況からゲームは始まる。まず,目の前で殺された仲間の恨みを晴らそう。そのためには「マンチー」という敵の正体を暴かなければならない。この「ロジャー号」の謎も解明しよう。「なぜ無人のまま放置されていたのか?」,「乗組員は?」,と謎は尽きない。こんな調子でプレイヤーは徐々に大きな謎へと挑戦していくのだ。

「世の中やっぱりゼニや」というわけで,生活するための仕事も決めなくては。「貿易商」,「賞金稼ぎ」,あるいは「海賊」。ただし,「貿易商と決めたらずっと貿易商」なわけでもない。職業は自分の心の中で決めておくだけ。自分のポリシーに従って,自分自身の行動を決めていくのだ。海賊をやりながら賞金稼ぎをやってもいいし,貿易商をやりながら賞金稼ぎをやってもいい。たとえば,「目の前にいるタンカーを攻撃するか,しないか」は,最初に心に決めた職業に関係なく,気の向くままである(あとのことは責任は持てないが……)。

X68000用 5″2HD版2枚組 9,800円(税別)
ウェーブトレイン/JICC ☎03(3239)0183

ゲームシステム

さて,まずは背景世界の説明から。ゲームを始めてすぐに,プレイヤーはある帝国統治下の宇宙空間にいる。ここにはいくつかの星系が存在していて,自由に移動することができる。ひとつの星系は,恒星,惑星,その他の小惑星や惑星間物質,人工建造物から構成され,主要なスターベースが数カ所ある。そこが,さまざまなイベント,情報収集等を行う拠点となる。そのほか,なんでもないような宇宙空間にも謎が隠されている場合もあるので要注意。

この世界には,いろんな宇宙人が生活している(なぜか言葉も同じ)。なかにはプレイヤーをだますやつもいるが,たいていはなんらかの情報を持っているので積極的に話してみるべし。

また,いろんな船が宇宙空間を航行している。タンカー,ハンター,帝国の誇る戦艦など。ぜひ近づいて見てみたい。

さて,プレイヤーの操作画面は,大きく分けて4つ。

1) 星系間ナビゲーションモード

現在いる星系の全体図を見ることができる。現在位置や惑星の位置の確認,空間物質の確認,ベースへの移動などを行う。自動航行システムも備えていて,星系内の移動は必ずこのモードで行う。

2) フライトシミュレーションモード

3Dポリゴン画面。敵機とのドッグファ

3D画面は結構地味

イトや，ベースにドッキングするときには このモードになる。手動で行うので失敗することもあり，アクション性が高い。

3) RPG画面

スターベースや基地の内部では，船を降りて自分の足で歩き回る。トップビュータイプの画面で，操作はきわめて簡単。4方向に動かして人に近づき，話をするだけで，ゲームが進行していくようになっている。会話もコマンド選択方式になっているので，それほど難しくはない。

4) HIVE！

「たいていの酒場には置いてあるアクションゲーム。帝国で大流行している」という設定らしい。RPG画面から，この「ゲーム内ゲーム」を遊ぶことができる。ただし，ストーリーの本筋には関係ない。こんなものがあるとは，実にアメリカンだ。

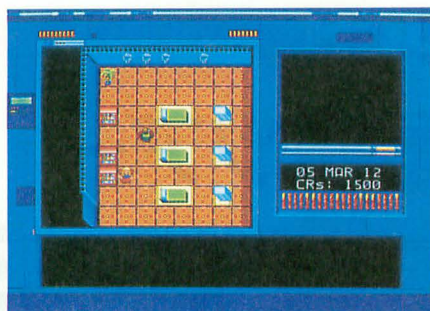
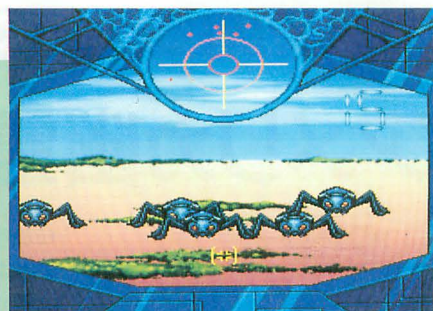
この4つのほか，星系間を移動するときに利用する「マリーゲート」がある。特定の2つの星系をつなぐかけ橋みたいなもの。ほかの星系に行くときは，必ずこれを通る。輪が連なってトンネル状になっていて，そこからはみ出さないように進まなければならない。はみ出すと，多大なダメージを受けたうえ，元の星系へ戻されてしまう。

一般的に，マリーゲートをくぐるときとかスターベースへドッキングするときは，かなり緊張する。その緊張が逆に面白さになっていて，シールドがなくなりかけたときに来た敵とは，マジになって戦ってしまった。

海賊稼業 ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

では，海賊生活の一部をお伝えしよう。

俺はまだペーペーの海賊だ。だが志は大きい。いつか帝国一になってやる。最近の戦いで俺の名声も少し上がったってもんよ。「厄介な」から「乱暴」になったぜ。この先どこまでレベルアップするか知らねえが，ハンターに狙われることも多くなったな。マンチーとか海賊の首には，帝国が賞金をかけてるんだ。貿易組合を敵に回した以上，

星系間移動はこのナビゲーションモードで

宇宙船を降りるとウルティマタイプの画面に

これが「HIVE！」，なかなか面白そう

いまさら貿易商にも戻れねえしなあ。平和な生活はできそうにないぜ。

ややっ！ 前方に見えるのは惑星だな。なになに「強力な重力源に接近中！」だと。あの重力から抜け出せなくて，燃え尽きちまったマヌケ野郎もいたっけな。普通の船には大気圏突入能力なんかないんだよ。

次は小惑星群か。「ガン！ガン！ガン！」痛てえ。こんなとこにいたら，船が壊れちまう。

おっと，そんなことしてる間にもうシールドがないな。ベースで補給するか。じゃ，ナビゲーションモードへ移行して，と。あばよ！

ゲーム総括 ◆◆◆◆◆◆◆◆◆◆◆◆◆

この「スペース・ローグ」，どこかのゲームに似ていると思わないだろうか。そう，あのアルシスの作った「スタークルーザー」（注1）だ。3Dポリゴンとかアクション性とか外見上似ているところは多い。根本的に違うのは，設計思想だ。「スタークルザー」はひと言でいえば「アクション映画」である。プレイヤーがやるべきことは最初からお膳立てされていて，アクションをしながらストーリーを傍観していればよかった。しかし，「スペース・ローグ」は「シミュレーション＋RPG」である。断片的なストーリーがそこらじゅうに散らばっている。プレイヤーは自分でそれを拾い集めて，組み立てなきゃならない。アクションをしながらストーリーを組み立てるのだ。どちらがいいとはいえないが，前者が日本的，後者がアメリカ的といえるだろう。

最後に，その「スペース・ローグ」の独断的評価を下してみたい。

●良い点

・適度な緊張感がある

ダメージ制で，ダメージを被る機会が意外に多い。ベースにドッキングするだけでも緊張する。疲れるといえば疲れるが。

・プレイヤーの行動の自由度が高い

謎の核心部分以外には，最初から行くことができる。最初にすべての星系を見ておくのもいいだろう。

・X68000版はグレードアップしている

「重要人物の顔のグラフィックが表示されるようになった」，「"HIVE！"が本格的になった」，「アナログジョイスティックに対応した」，以上3点。

・登場人物の居場所がランダム（一部固定なのもいる）なので，一度終わっても何回でも楽しめる

●良くない点

・マリーゲートなど，8ビットでもできそうなグラフィックが見受けられる

・コクピットのグラフィックは「スタークルーザー」を見てしまうと見劣りがする。積み荷や装備が見にくいなあ

移植元（もともとはAppleII用）があるのでしかたないが，全体的にX68000のパワーを使い切っているとは思えない。

注1：この「スタークルーザー」は，X1用ゲーム全部の中で，5本の指に入るといっても過言ではない。デザイン，ストーリー，アクション，ともに素晴らしい完成度でいまでも十分通用する。X68000へも移植された。

3Dポリゴンにばんざい！

そういえば音楽のことを書かなかった。全編にわたって，小気味いいテンポの曲が流れている。タイトルバックの妙な曲は，結構耳について離れない。

古臭いなどと書いてしまったが，逆にそれだけ昔から3D技術がアメリカにはあったのだ。日本ではいまだに，これだけ動く3Dポリゴンのゲームは少ないだろう。しかもシミュレーションとRPGまで楽しめるのだ。X68000のパワーうんぬんの前に，ソフト技術を評価するべきかもしれない。

総合評価	0　　5　　10
アクション	★★★★★★★★
RPG	★★★★★★★
シミュレーション	★★★★★★
音楽	★★★★★★★
操作性	★★★★★★★★★
アメリカン	★★★★★★★★

アメリカ幽霊屋敷ツアー

1920年代，アメリカ，ニューカムというような名の街とくると思い出すのは……，ラブクラフト？　いやいや，違います。昔，他機種で出ていて，ようやくX68000にも移植された「ラプラスの魔」の話なんですよね。

うう，なつかしいったらありゃしない◆◆

この，ラプラスの魔というゲーム，今のゲームとしてもそうなんですが，その昔，他機種で発売されていた当時から非常に変わったゲームシステムで知られてたRPGなんですよね。

何が変わってるかというと，まずモンスターを倒すだけでは全然お金を手に入れることができないので，金を手に入れるためには，幽霊屋敷に入り込んで化け物どもの写真を撮ってそれを街で売り飛ばして金を手に入れなくてはいけないとか（逆にいうと今も昔もRPGってモンスターを殺しまくりの金奪いまくり，のゲームだったわけだな），ヒットポイントがひとつじゃなくて生命力のパラメータと精神力のパラメータの2つのパラメータに分かれてるとか（生命力のヒットポイントがなくなると死んでしまうってことは精神力のヒットポイントがなくなると，心が死んじゃう……つまりが発狂しちまうわけだ）とか，ほかにもいろいろな技能を金で買ってこなきゃいかんとか，作ったキャラの顔写真がどいつもこいつも不気味だなあとか（本っ当に不気味でさあ，女のキャラって作る気がしなかったんだよね……）。

さてさて，時は192X年，アメリカ，マサチューセッツはニューカムという街。ジャーナリスト，探偵，それに霊能者たちといったゴーストハンターたちが，幽霊屋敷と評判のウェザートップの館に挑んでゆきます。さぁ，彼らに明日はあるのか⁉

お知らせ

1月号の付録ディスクで配布したウイルス検出プログラムDOCTOR 2．Xに不備がありラプラスの魔のディスクに対して不適当なメッセージを表示することがわかりました。皆さんに不要なご心配をお掛けしたことをお詫びいたします。

なお，DOCTOR2．Xを組み込んでいる方は，ラプラスの魔を起動の際にOPT．1キーを押しながら立ち上げるか，170ページの方法でDOCTOR2.Xのデバッグを行ってください。

X68000用　5″2HD版3枚組　8,700円(税別)
エム・エー・シー ハミングバードソフト
☎06(315)8255

免税店はないけれど◆◆◆◆◆◆◆◆◆

え，みなさま右手をご覧ください。いちばん高いのが中指でございます（ああっ，古い……）。コホン，え，失礼いたしました。みなさま，本日はわがラプラス観光のアメリカはニューカム幽霊屋敷ツアーにご参加いただき誠にありがとうございます。

ただいまご覧になれますのがやっと辿り着きましたニューカムの街です。これよりお待ちかねの免税店……ではなくて，街での買い物にまいります。ちなみにヴィトンもグッチもありませんのであしからず。

はい，みなさま，左手をご覧くださいませ。こちらがニューカム名物，なんでも治す奇跡の教会でございます。もし屋敷内で負傷されたり，毒を受けても，マヒしても，挙句の果てに死んじゃおうが，（ピー）になってしまおうが，この教会で手当をうければたちどころに治ってしまいますので，みなさま安心して死んでくださいませ。なお，体力と精神力の回復だけでしたら無料ですが，死んでしまったり発狂されてしまった場合は，はっきりいって大量のお金が必要ですのでみなさま，死ぬときには十分なお金を用意してお死にくださいませ。地獄の沙汰も金次第ぃ，ほっほっほっ。

さて，右手にご覧になれますのが武器屋に道具屋，それに占い師の店でございます。

まず武器屋でございますが，リボルバーや弾，サーベルなど，よろず武器を扱ってございます。

続いての道具屋でございますが，こちらは生命力の回復をするための医療箱，精神力回復用の護符，それに精神攻撃に使う機械などがおいてございます。

また，当幽霊屋敷の幽霊ならびにモンスターたちは，ビンボービンボー涙のビンボーなのでございましてほとんどの場合倒しても金目のものは持っておりません。したがって，ジャーナリストの方をパーティにお連れして，この道具屋でカメラとフィルムをご購入のうえ，戦闘中に幽霊どもの写真をバシバシ撮って，ホテルへ帰って売り飛ばしてくださいませ。ついでにそやつらをぶっ殺してしまえば経験値も稼げるし一石二鳥でございましょう。はっはっはっ。

続いては向かって左側，占い師の店でございます。ここではレベルアップとスキル

を買えます。みなさまが幽霊屋敷で化け物を倒したあとでここに参りますと，レベルアップで体力と精神力の最大値を上げることができます。スキルというのは，早い話“腕”でございます。たとえば，写真のスキルがあれば写真がうまくとれる，というわけです。またほかにも銃を使うためのスキル，格闘のスキルなど生き残るための能力はここでかなり左右されてまいります。地獄の沙汰もスキル次第というわけですね。

それではみなさま，買い物時間にいたしますので，たっぷりお楽しみくださいませ。

恐怖の幽霊屋敷ツアー◆◆◆◆◆◆◆

さて，これよりバスは街外れにありますウェザートップ館へと向かいます。これからが本番というわけですね。ふへへへへ。

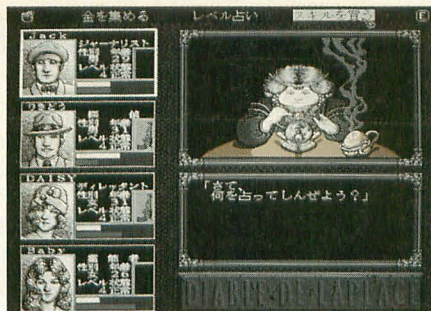
右手にご覧いただけるのが，かの有名なウェザートップの幽霊屋敷でございます。この館は，ただの荒れ果てた無人の館だったのでございますが，最近，子供のバラバラ死体や無惨に喰いちぎられた死体が庭先で発見されましてから一躍有名になりましたのでございます。これよりみなさまは屋敷の中の冒険へと参られるわけです。

みなさま，みなさま押しあわずに落ち着いて一列に屋敷へお入りくださいませ。おや，お客様ドアを引っ張ってどうなさったんですか？　残念ながらただいまみなさまが入られた扉は，入ったとたん鍵が閉められてしまいまして……。入ったら最後，鐘つき堂の鐘を鳴らさないと扉は開かないようになっているのでございますよ。ふははははは（←デーモン小暮調に読むこと）。

さて，さっそく屋敷に入りましたところでみなさま右手をご覧くださいませ。右手に見えますのがみなさまの前に出発して，全滅したパーティご一行様の変わり果てたお姿，はっきりいって亡霊でございます。みなさまこのようなお姿になられませんようご注意くださいませ……。

さて，これより半日観光……じゃなくて，まず屋敷の中をてっぺんにあります鐘つき堂へまいります。おっと，さっそくのモンスター，スケルトンのおでましでございます。みなさま武器およびカメラのご用意はよろしいでしょうか……？

やっぱりX68000だもんね◆◆◆◆◆◆◆

さてさて，屋敷にあるいは地下に入るといろんなモンスターたちが相手をしてくれます。いや，本当にいろんなモンスターがいるもんですねえ。ねずみとかクモとかスケルトンやらグール，挙句の果てには不気味な青い目の人形さんまで。そやつらが大挙して押し寄せてくるさまなんかは，ほとんどホラー映画です。さらに敵ばかりでなく助けを求めてくる幽霊なんてのまでいますから，そういう人（すでに人ではないってか？）に会った場合には，話をちゃんと聞いてあげましょうね。聞くだけで経験値がもらえますから。

あ，そうだ，敵といえば，敵が降参してきたらたーっぷり，ネチネチと尋問してあげましょう。たまーに物をくれたり，ベラベラと秘密をしゃべってくれるみたいです，はい（……にしてもクモやねずみの写真でお金をくれるというのは……クモやねずみが珍しいのかな，アメリカじゃ）。

このゲームが8ビット機で出た頃，実は私もちょこっとだけやったんですよね。昔はメモリも少なかったからディスクアクセスがやたらと多くて，時間がかかって非常にうっとうしかったんですよね（まあ，あの頃はそれが当たり前だったんだけど）。

ところがX68000ではさすがにオンメモリというわけにはいきませんけど，それでもちゃーんと読めるだけメモリに読み込んでいるので，余分なディスクアクセスがなく非常に快適です(もっとも私はメモリ2Mバイトにしているのでそのせいもあるのかな……)。それに，グラフィックも全部書き換え，それもタイリングペイントじゃなくてちゃーんと中間色してます。えらいえらい。グラフィックのセンスもかなり当時に比べると向上してるようで，まだ男の顔は不細工ですけど，女のキャラはまともに見られる顔になりましたしね。写真をよく見てもらえればわかると思うけど，私の作った“りん”てキャラクターなんですけど……妖しく美しくっていいと思いません？　ま，ほかのマシンとは一線を画すスーパーAVマシン，X68000用に移植するんだから，このくらいバージョンアップしても当たり前っていっちゃなんだけど，当たり前のことを当たり前にやってくれるのは非常に気分のよいことなのであります。

ただ，ちょっと不満をいわせてもらえば……このゲーム，ちょっと，難しいよー。すぐ，全滅しちゃうんだもん。私はおかげで写真撮影中にキャラクターが全滅しちゃってえらい苦労したんだぞぉ……，うるうる。でも悲しいといえば，何度キャラクターが殺されて全滅させられても，結局新しいパーティを組んで幽霊屋敷に送り出してしまう自分がいちばん悲しい……。

最後に，みなさま本日はアメリカはニューカム幽霊屋敷ツアーにご参加いただき誠にありがとうございました。右上にご覧になれますのが，みなさまの前に出発して全滅したたパーティご一行様の変わり果てたお姿でございます……。

おばちゃん，スキルひとつね！（駄菓子屋かっ）

幽霊さんも人生相談があればねえ……

リフレッシュされてます

うーむ。ゲーム自体は非常に古い物なんだけど，X68000に合わせてしっかりとシステムが新しくなっていることは大変よいことだと思います。でも，どうせならゲーム自体新しくX68000オリジナルでやってくれてもよかったのではないかとも思うのですが……。ま，グラフィックもちゃんとX68000してる（ちゃんとX68000に合わせてグラフィックを描き直すっていうのは本当は当たり前じゃなくちゃいけないんですけどねえ）し，プルダウンメニューしてるし，なによりきっちりよいプログラムを組んで快適な操作感を提供してくれているので……よしとしますかね。うん。

ところで，非常に個人的な話になるんですが。このゲームの広告には“次はゴーストハンター2で会いましょう”ということになっているんですが……。できればロードス島戦記でお会いしたいなー，なんて思うんですが，どんなもんでしょ，ハミングバードソフト様(間違えても，次は地獄の練習問題なんていうんじゃありませんように……)。

総合評価	0　　5　　10
拡張メモリ対応	★★★★★★★
さまよえる手	★★★★★
作ったキャラクターの顔	★★★★★
プルダウンメニュー	★★★★★★★
ゲーム難易度	★★★★★★★★★
おすすめ度	★★★★★★

闇の中で眠れ

Ogikubo Kei
荻窪 圭

ひと月お休みをもらって，またまたカオスの逆襲の登場。どっぷりとはまっている荻窪氏ですが，先々月とはうって変わって至極マジメでカタい文章に，このゲームのシビアさが伝わってくるようです。

自由業なのをいいことに，真っ昼間からずっとテレビにかじりついていた。銀座でインタビューされた人が，戦争によって何が心配かと聞かれて，原油や物価の高騰だと答えていた。ほかに心配することはないのだろうか。前線の軍人さんの命が心配とでも答えればまだかわいげがある。

日本はどうすべきかと聞かれた善良な市民は，モーニングに載っている本宮ひろしの漫画と同じようなことを真面目な顔でいっていた。日本はどうすべきか。こたつに入ってみかんでも食べていればいいではないか。それで自分たちの平和ぼけでもかみしめていればいい。

私はといえば，ステルス戦闘機を見ながら，ダンジョンをさまよっている。

ダンジョン・マスター。このゲームは古今東西に類を見ないほど，真面目で硬派なゲームだ。まったく遊びはなく，アイテムも敵もダンジョンもハードボイルドだ。ボタンがバカになったマウス・トラックボールは買い換えられ，私はダンジョンを走り，止まり，眺め，考える。悲しむ者もなく，正義は勝つという運命もない。この冷たさだけが，危険を嗅ぎ分ける。

だから，今回のレビューはそんなダンジョンに敬意を表して，堅く，堅く，淡々と進められる。そこにあるのは，アウトラインであり，ディテールは実際に歩いたものにしかわからない。勝利のあとに見出せるのは，ロード・カオスのパラノイアに最後まで付き合わされた虚しさだけだ。

最終的に世界が平和になったとしても，戦いの虚しさは晴らされない。新しく見つけた武器のテストのために殺された敵も，勝利のためにやむをえず殺された敵も，すべては虚しさの前に等価であり，平和もまた人間の根底を流れるパラノイアックで独りよがりで傲慢なロマンにかぶせられた閉じ蓋にすぎないのだ。

4◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

ダンジョンは4つに分かれ，合流し，再び4つに分岐して最上階で合流する。

最初の分岐点を単に「分岐点」と呼ぶ。ここはプレッシャープレートがあるひとマスの小さな部屋であり，ワープゾーンからワープによって，あるいはその真上にある穴から落ちて辿りつくことになる。スタート地点からこの分岐点に来るのは実に簡単で，早ければ穴に落ちてワープすればすぐだ。しかし，パリに買い物にいくHanakoの精神によるなら，分岐点までのささやかな道のりで，戦争準備を整えるべきである。

分岐点から東西南北へ道が分かれる。

その道には進行方向に対してのみ壁が開かれるようなしかけが施されている。

4つの道は次のように名づけられている。

“KU”，“DAIN”，“ROS”，“NETA”。

それぞれ職業でいい換えればファイター，ウィザード，ニンジャ，プリースト，日本語では戦士，魔術師，忍者，僧侶だ。

それぞれの道にふさわしい怪奇なダンジョンをさまよい続けると（不用意に穴に落ちたり，いくつかの隠された通路を通って分岐点へ戻ったりしない限り） 2つめの分岐点へと辿りつく。ここに辿りつくまでが最初の難関。頭に入れておいたほうがいい流れがある。降りて，昇れ。2つめの分岐点は，分岐点より上にある。

2つめの分岐点には看板がかかっており，名前がつけられている。“DIABOLICAL DEMON DIRECTOR”，略して“DDD”，訳して“悪魔の支配領域”である。以降，DDDと呼ぶ。どの道を通って来ても，このDDDの入口に辿りつく。

DDDの看板横にある階段を上ると，悪魔が待っている。そこがDDDだ。

DDDは複雑であり，そこから再びKU, DAIN, ROS, NETAの4つに道が分かれる。それらの入口を見つけることは極めて困難だ。道を塞ぐピットを閉じるために運と知恵を必要とするだろう。DDDをすりぬける快感は味わった者にしかわからない。

どの道を進んでも，どこかに“コーバム鉱石”がある。コーバムがカオス・ストライク・バックの目的だ。

コーバムはピットに囲まれた部屋の中央の柱，4面にそれぞれ収められている。コーバムを入手するのは困難を極める。息を止め，素早くパーティを動かす。

手にしたコーバムは最上階（だと思われる）にある“フル・ヤの炉”に投げ込むことになっている。文字どおり，投げ込む。

フル・ヤの炉に至る部屋で4つの道は再び合流する。ロード・カオスやうっとうしい悪魔，ややこしい罠に囲まれた大きな部屋。プレイヤーはここで迷い，ピットに落ち，嘆き，さまようだろう。答えはひとつ

X68000用 5"2HD版2枚組 9,800円(税別)
ビクター音楽産業 ☎03(3423)7901

24kHzモードで遊ぶ

カオス・ストライク・バックは24kHzモードで遊ぶことができる。これは有名な話。

その実現はゲームディスクの設定データを変更することによって行われる。SHIFTキーを押しながら起動すると環境設定メニューが立ち上がり，そこで24kHzモードを選ぶと，ディスクにその設定が書き込まれる。よって，プロテクトシールを一度はがす必要がある。一度設定すれば次からは，そのモードで立ち上がるので，プロテクトシールは貼り直して構わない。

私は24kHzで遊んだが，どちらがよりいいということは感じない。どちらでもよく似たものである。今回の画面写真はみな24kHzモードで撮ってみた。

しかない。

4つのコーバムをフル・ヤの炉へ投げ込むと，終わる。ロード・カオスと無理に戦うことはない。

KU

カオス・ストライク・バックでは「どこまで行った？」「えーっと，地下7階」という会話は成り立たない。いたるところで分岐しており，「どこまでいった？」と尋ねられて正しく答えるにはいろいろと複雑な解答を用意する必要がある。

目安となるのは，DDDへ辿りついたか，コーバムをいくつ手に入れたかであるが，分岐点からひとつの道を通ってDDDに至り，そこからまたがんばってコーバムをフル・ヤの炉に投げ込む，という猪突猛進を4回繰り返す人もいれば，分岐点からDDDへの道を4つとも歩き回って自分たちを鍛えアイテムも十分手に入れたところで，あらためてコーバムへ挑戦する人もいるだろう。

ここでどの道がいいかと尋ねられれば，KUがなかなか楽しい，と答えておく。謎がそれほど複雑ではないからだ。

しかし，堅くて厄介なデス・ナイトうようよ。更にDDDの先へ行ったりすると，ドラゴン巣窟なんていうレッドドラゴンうじゃうじゃ。ああ嬉しい。ここでドラゴンを退治したら箱に肉を詰め，DDDの階段の下にでも置いておくべきだ。どの道からでも必ず通過する(というより連れてこられる)ので，キャンプ地にはもってこいだ。KUで食料を集めておけば餓死することもない。

DAIN

今回のダンジョンはあまりにも複雑。何階にいるかもわからない。マッピングに頼ると時間ばかりくってしまう。重要なのは自分が何階にいるかではなく，各ポイントに対して相対的な高さをつかむことだ。

今回はゲームが難しいぶん，マニュアルでサポートしている。牢屋からチャンピオンを選ぶ方法やそのマップ，スタート地

分岐点を上から見たところ

DDDの看板

フル・ヤの炉

点の切り抜け方など，わかりやすい。新しく付け加えられた魔法の地図用魔法（これは僧侶の魔法だ）以外の魔法表もついてくる。見ながら魔法を唱える。楽である。

今回は前作によく見られた看板によるなぞなぞは減り，より複雑な，壁でない壁に隠された通路やその奥のスイッチ，間欠開閉穴などアクション性の濃いいやらしいものになっている。

今回の最大の罠は魔法の地図かもしれない。地図に魔法をかけると壁でない壁がわかってしまうからだ。魔法の地図に頼るあまり，壁の小さなスイッチを見逃してしまったり，アイテムを拾い損ねたり，重要でない場所にとらわれたりしないよう注意する必要がある。あくまでも魔法の地図は旅のサポートにすぎない。

ROS

適当に道しるべを置きながらダンジョンを進む。いらないアイテムを置くのもいいが，本当に重要な，未来へ通じる道にはゾーキャスラーだ。強い魔術師がいる限り無限に作ることができ，光って目立つから。

魔術師はどんどん強くなっていく。

戦士もどんどん力を蓄えていく。

僧侶も同様。

忍者の能力はそれほどアップする要因がないが，優秀な忍者がいなければならないわけではない。ボウガンでもあればよい。

投げてスイッチを押す。これは快感だ。

戦いを重要視しないROSやNETAの道で重要なのは，プレイヤーの柔軟な頭脳であり，あきらめず挑戦する心である。どの道でも重要なのは肉弾戦に強い戦士と強いファイアーボールとポイズンフォウを撃てる魔術師だ。そして，知恵。知恵なくしてカオス・ストライク・バックは終わらない。

NETA

一番ユニークなNETAの道。モンスターをコントロールしてプレッシャープレートを踏ませないと先へ進めない部屋は秀逸だ。むやみに殺しても駄目なNETAの道である。

モノを置けば反応するスイッチ，パーティが乗らねば反応しないスイッチ，モンスターにのみ反応するスイッチ。プレッシャープレートにも種類がある。

そして，どの道もフル・ヤの炉に通じる。

ひとつの道を終えるたびに分岐点（すでにガラクタ置き場兼食料庫としてジャンクショップ並みの乱雑さだ)でほっと休養する。

通れる壁・小さな壁のスイッチ・人が乗ると穴の開く床・スイッチを押して数秒だけ閉じるピット。つまり，床と壁。

前作のときと同じようなことをまたいおう。探検を終えたダンジョンには，あなたが開けられなかった扉の数だけ物語があり，あなたが落ちたピットの数だけ物語がある。残るのは物語である。英雄ではない。

ダイハードとダイハード2

懸念された音楽も入っていなかった。ポピュラスの心臓音のような音がときどき鳴るのが気になるが，これは，サービスである。重要な地点に来たことや，モンスターが近くに寄っていること，隠れたスイッチや通れる壁があることを教えてくれるサービスなのだ。

第1作は衝撃的でかつ面白かった。第2作はよりスケールアップしており，面白くはあるが衝撃的ではなかった。なんの話かというと，「ダイハード」と「ダイハード2」なのだが，ダンジョン・マスターにも似たことはいえる。

ただ，映画とは異なり，ゲームではその世界への順応という過程が必要なので，前作でダンジョン環境に順応したプレイヤーにとって，カオス・ストライク・バックのほうがゲームをより本質的に楽しめるということはいえるかもしれない。

前作より難しいとはいっても，中だるみしない構成やビジュアルやサウンドの演出力は秀逸である。私はやはりダンジョン・マスターシリーズが（ダイハードよりも）好きだ。

総合評価 2	0　　　5　　　10
マニュアルの親切さ	★★★★★★★★★
落し穴の多さ	★★★★★★★★★★
探検のわくわくさ	★★★★★★★★

AFTER REVIEW

今月はシステムサコムの「闇の血族」です。ノベルウェアというところで賛否両論となりそうですが，どうでしょうか。たしかに，こういうものも質さえ高ければ楽しめそうですよね。この作品の場合はどうかな？

闇の血族

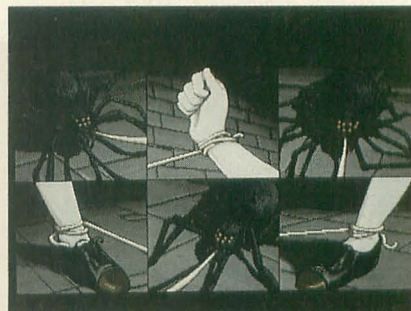
▶なかなか楽しかったです。72点ほどでしょう。いちばん驚いたのはデータ圧縮率です。とてもディスク3枚とは思えませんねえ。ノベルウェアのシステムもだいぶ高くなってきて使えるレベルになってきたのではないかと思います。音楽もとてもいいと思います。ただ，私はMIDIを持っていませんから，ウワサの映画館のような……というところまではいきませんでしたが。

しかし，不満点も3つほどあります。ひとつはキャラクターデザイン。なにもプロを使えとはいいませんが，年齢設定と顔のギャップがすごい。里沙なんてほとんどオバサンだ。

もうひとつは，次回予告。なんであそこまで凝る必要があるのか，理解に苦しむ。あの分のデータ量を少しでも多く本編のほうに割りふってほしかった。あれじゃまるで，無理やり盛り上げて完結編を意地でも買わせようとしているみたいである。

最後のひとつは価格設定。8,800円×2は絶対に高いと思う。せめて，6,000円×2か，10,000円×1が普通だと思う。それと，このノベルウェアシステムはまだまだビジュアル性に欠けており，小説を読まない，あるいは読めない人からはかなり敬遠されています。フロッピーディスク（1Mバイト）というメディアでは限界なのかもしれませんが，もう少しシステムを見直したほうがいいと思います。　泉　哲也　岩手県

▶闇の血族については皆さん意見が厳しいですね。私はひさびさに満足できるゲームだと思いましたが……。演出なんかはかなり凝っていますし……。ちなみに私は新井素子さんのファンです。「はふ」，このことばにつきますね。しかし，原作が男とは……。　真鍋　博之(19)東京都

▶最初に広告を見たときは，なんとなくいやだなあと思った。だって，あの絵でしょ。

魅由が蜘蛛に襲われるところ。連続写真でどうぞ

ああいうタイプの絵は苦手なんですよ。でも，マヤ文明の建物とかが出ているのとか，「南米の血に隠された秘密とは？」とかいうキャッチコピーが広告に出ているのを見て，パソコンショップに行ったときに衝動買いしてしまったのです。家に帰ってゲームをやり始めると，なんか勝手に進んじゃってなかなか自分の意思が反映されない。しかも，私の目的とは違って普通の部屋の中とか，喫茶店とかの日常的な風景の連続でなんとなく欲求不満。前編は予告編がいちばん出来がいいかな，って感じです。それで完結編のほうはというと（途中までを見ると絶対に最後まで見ないと気がすまないもんで），最後のほうのアニメーション，アレはよかった。「バーン，バーン，バーン」とかいう音とシンクロして絵が出たりして，ワクワクしました。　谷口拓哉(26)群馬県

▶全然進めないアドベンチャーゲームと，なにもしなくても終わってしまうアドベンチャーゲーム。どっちがいいんでしょうねえ。ところで，実はお金がなくて完結編は買っていないのですが，結末が気になってしょうがありません。どうにかしてください。　和田篤(16)島根県

▶私も完結編をやってみたぞ。2時間ぐらいで楽にやり終えちゃった。コバルト文庫のような文体，明るいようでちょっと暗いグラフィック，メキシコ神話に題材をとったヘビーなストーリーと，なかなか楽しめた。だけど，これをアドベンチャーだと思うと，プレイヤーの思いどおりにならない魅由に「はふ，ちょっとため息（Ⓒシステムサコム）」。このゲームは変に感情移入しないで，ストーリーを読むつもりで楽しむのがいいのだ。まだノベルウェアならではのメリットを生かしきれてないように思うけど，こういう斬新な試みで頑張るシステムサコムは応援したいな。　(浦)

▶私が思うにこのゲームの魅力は次の3点にあると思う。まず，ひと味違うグラフィック。独特の雰囲気があるのだ。そのせいか，広告の「美少女探偵魅由」というのが適切がどうかが議論を呼んでいる。次に音楽。内蔵音楽，MIDI共にグレードの高い音質でBGMが奏でられるのだ。MT-32使用時でもプリセット音色のみならず，オリジナル音色も使っているようで新鮮な感じだ。曲のほうもメインテーマのバリエーションを各曲に盛り込んだりしていて，映画音楽のノリ。あとはなんといってもメッセージ。

魅由が空中浮遊でいたぶられる

「んーもう，JESUS」，「はふ，ちょっとためいき」，「ワタシデナイワタシ」などは流行語になったほどだ（反論は却下）。「第4のユニット」のようにシリーズ化してほしいな。はふ。　最近溜息の多い(善)

予告：来月は遊撃王IIとガンシップでもやろうかなと思っています。皆さんどんどん感想や意見をお寄せください（特に，長めの文章を望みます）。

発売中のソフト

★Misty Vol. 7　データウエスト
XIturbo用　5"2D版　5,000円(税別)
X68000用　5"2HD版　5,000円(税別)

★中華大仙　シャープ
X68000用　5"2HD版　7,800円(税別)

★Musicstudio PRO-68K ver. 2. 0　シャープ
X68000用　5"2HD版2枚組　28,800円(税別)

★びんびん麻雀ピーチエンジェル　データ集
ブラザー工業（TAKERU）
X68000用　5"2HD版　2,000円(税別)

★アルガーナ　ブラザー工業（TAKERU）
X68000用　5"2HD版　6,800円(税別)

★ブルトン・レイ　シナリオエディタ
システムソフト
X68000用　5"2HD版2枚組　5,800円(税別)

新作情報

★DRAKKHEN　EPIC/SONY RECORDS
X68000用　5"2HD版　価格未定

★マーブルマッドネス　ホームデータ
X68000用　5"2HD版　価格未定

★ファンタジーIV　スタークラフト
X68000用　5"2HD版　9,800円(税別)

★エイリアンシンドローム　電波新聞社
X68000用　5"2HD版　価格未定

★プリンス・オブ・ペルシャ　ブロダーバンドジャパン
X68000用　5"2HD版　価格未定

★パロディウスだ！　コナミ
X68000用　5"2HD版　価格未定

★生中継68　コナミ
X68000用　5"2HD版　価格未定

★ノスタルジア　タケル
X68000用　5"2HD版　11,800円(税別)

★ボンバーマン　システムソフト
X68000用　5"2HD版　7,800円(税別)

★ファランクス　ズーム
X68000用　5"2HD版　価格未定

★メルヘンメイズ　SPS
X68000用　5"2HD版2枚組　8,800円(税別)

★クォータースタッフ　スタークラフト
X68000用　5"2HD版　価格未定

★シムアース　イマジニア
X68000用　5"2HD版　12,800円(税別)

★ビーストロード　ホビージャパン
X68000用　5"2HD版　価格未定

★マーキュリー　マキシマ
X68000用　5"2HD版　8,800円(税別)

★シグナトリー　NCS
X68000用　5"2HD版5枚組　12,800円(税別)

★パワーモンガー　イマジニア
X68000用　5"2HD版　12,800円(税別)

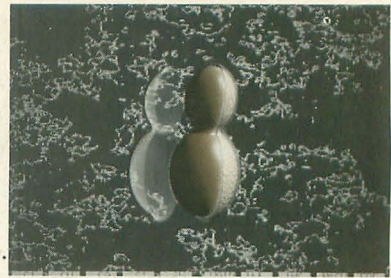
★シムシティー　テレインエディタ　イマジニア
X68000用　5"2HD版　4,800円(税別)

★サブナック　工画堂スタジオ
X68000用　5"2HD版2枚組　7,800円(税別)

★CARD PRO-68K ver. 2. 0　シャープ
X68000用　5"2HD版　価格未定

メタボールを君の手に

Tan Akihiko
丹　明彦

レイトレーシングではお馴染みのC-TRACEの最新バージョンがC-TRACE＋だ。これまでのver.3.0の機能に加えてさまざまな拡張が加えられている。目玉はやはりメタボール。パーソナルでメタボールが扱える日がきたのだ。

暮れももう押し詰まったある日のこと。今年はこれが最後かなと思いつつ，年末進行でいつもより早く仕事の片づいた編集部に顔を出したら，C-TRACE＋のサンプルがきているとのこと。もうちょっと早ければ先月号のグラフィック特集で紹介することができたのに，と思ったが，そのおかげで試用期間が1カ月もとれることになった。これで正月はメタボール三昧だぜ。

たいへん長らくお待たせいたしました，とでもいうべきなのか，待望のC-TRACE＋がついにX68000でも動き出した（PC-9801シリーズにはかなり前からあった）。

C-TRACE+とは

先月号ですでに予備知識もできていることだろうが，C-TRACEはレイトレーシングソフトウェア。レイトレーシングとは，3次元コンピュータグラフィックの描画アルゴリズムのひとつ。

3次元空間の中にいろいろな物体を置く。そしてそれらを写真に撮ると考える。カメラのファインダーに映った像ができあがりの画像にあたるわけだ。構図が決まればカメラの向きや場所も決まる。そこから本計算に入る。画面上のドットに対応する視線をドットの個数と同じ本数だけ発生させ，その視線が最初にぶつかったプリミティブ（基本立体）の色を調べて画面にその色のドットを打つ。これを画面上のすべてのドットに対して繰り返し，1枚の画像を作り上げていく。それがレイトレーシングだ。

そして今回発売されるC-TRACE＋は，C-TRACEの上位バージョンとでもいうべきものだ。C-TRACE「＋」というだけのことはあって，C-TRACEからずいぶんと拡張されている。なかでも目玉はメタボール。

X68000用　5″2HD版　198,000円(税別)
キャスト　☎03(3705)1065

メタボール

メタボールは不思議なプリミティブ。日常的な感覚でその性質を捉えるのは困難であろう。

メタボールの「メタ」は，変形を意味するmetamorphosisからきているという。そう，メタボールは変形するプリミティブなのだ。ふだんは球（または楕円体）なのだが，ほかのメタボールが近くにくると，干渉しあって変形，融合などの現象を引き起こす。その変形の具合をうまくコントロールして，目的の形を作り上げる。思いどおりの形を作るのが難しい，という意味での制限つきではあるが，メタボールは自由曲面を実現するひとつのアプローチなのである。

●メタボールとは

レイトレーシングでよく使われるプリミティブ（球や楕円体など）は，その表面を境として内側が1，外側が0，というように，いわばデジタルにできている。

これに対して，メタボールは，中心が2.0，その値は中心から少しずつ外側にいくにしたがって小さくなっていき，しまいには0.0 になる。そして，1.0以上の部分だけが目に見える，つまり1になる。ほかは見えない（つまり0）。初めはアナログ（2.0〜0.0）で考え，最後にデジタル（1か0）にするというわけだ。もちろん，ここで挙げた数値（2.0や1.0）は例として挙げただけで，ほかの値を使っても構わない。

これを，空間に分布する濃度としてイメージしてみよう。メタボールの中心部の濃度は最大，中心から離れると濃度は少しずつ下がる。その濃度をある一定の値で切り，濃度がその値より大きいか小さいかでその部分がメタボールの内側か外側かを決めることにする。

中心部の濃度を，メタボールの重みという意味でウエイト，外側か内側かを決める境界の値をしきい値と呼ぶ。

変形の様子はメタボールの半径やウエイト（重み），それにメタボール間の距離などによっていろいろと変化する。たとえば，メタボール間の間隔が狭いほど変形は大きくなる。

●メタボールの変形

メタボールの濃度は，その中心からの距離で決まる。だから，メタボールが1個だけある状態では，常に球にしかならない。それを変形させるには，もうひとつ（またはそれ以上）のメタボールを使う。

メタボールは，単独では単なる球だが，その周りには0でない濃度が分布している（見えないだけ）。いわば，メタボールの周りには「場」ができているのだ（“場”とは電場や重力場といったときに使う“場”のこと。メタボールの概念は，電場のイメージに近い）。そこにほかのメタボールが接近する。そのメタボールも，当然周囲に場を作っている。両者の場は重なり合う。2つの場を足したものが新しい場になる。当然その新しい場は元の場とは形が違う。しき

い値を超える部分の形も変わる。こうして，メタボールは変形する。もちろん，3つ以上のメタボールの場合でも互いに干渉しあい，変形する。

面白いのはここからだ。場はなにもポジティブでなくたって構わない。ネガティブな場を作り出すメタボールがあってもいい。

負の重みを持つメタボールの周りには，負の場ができる。負ということは，それ自身の濃度は決してしきい値を超えることはない。それゆえ，負のメタボールは単独では見ることはできない。ところが，ほかの（正の重みを持ち，目にも見えている）メタボールのそばでは，負のメタボールは相手の場の一部を吸い取ってしまう。相手のメタボールの濃度が下がることになる。すると，いままでしきい値を超えていた部分が超えなくなってしまうのだ。結果として相手のメタボールはへこみ，ときには穴が開いてしまうこともある。

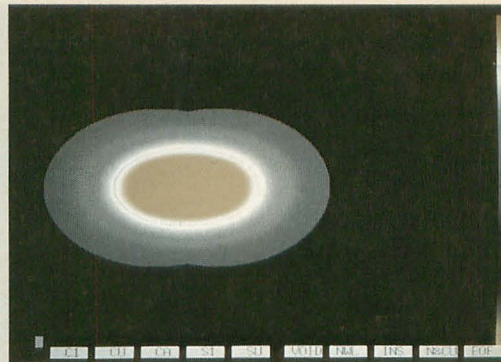
ドーナツ形も，たった2個のメタボールで作れてしまうのだ。

●視覚的に理解しよう

正のメタボール同士，また正と負のメタボールを距離を変えて置いてみた場合の場の変化を図示してみた。この写真の見方は，

・図の中の色は，物体そのものの色とは無関係である。これは，画面内の各点での濃度を表している。

・画面右端のカラフルな帯は，どの色がどのくらいの濃度であるかという凡例である。

それぞれのメタボールは濃度の広がった空間として表される。これは正負のメタボールによる変形のようす

中央（黒い線が入っている）を境目として，上が正の濃度，下が負の濃度を表す。中央（黒）は濃度0である。

・この図はあくまでメタボールの概念図であり，メタボールを直接表現しているものではない。第一，平面である。

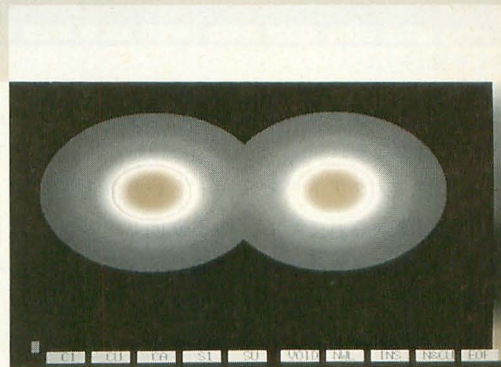
・白線はしきい値を表す。帯の，白線より上の部分に相当する色は見え，下の部分は見えない。また，白線そのものは物体の境界を表す。なお，線が乱れているのは，計算誤差で，本当は滑らかな曲線である。

といったところか。なお，この表示プログラムは僕がでっち上げたものなので，モデリングに使えないかとか，間違っても考えないように。しつこいが，平面である。

冗談はさておき，写真から，正のメタボール同士は干渉・融合してヒョウタン形になり，正と負のメタボールは相殺して穴が開いてしまうのがおわかりいただけることだろう。

この図では2つしかメタボールを使っていないが，メタボールはいくつでも並べることができ，それらはすべて互いに干渉しあい，複雑な変形をしてくれる。3次元空間では，ちょっと考えただけでは想像もつかないような面白い形ができることもある。

メタボールのグループ化と論理演算

これまでのレイトレーシングでは，用意されたプリミティブ以外の形を作ろうとすると，積み木のようにつなぎ合わせていくか，または論理演算を使ってプリミティブの一部を削り取るかしていた。一般的なレイトレーシング画像がレイトレーシングで作ったものだとすぐに見破られてしまうのは，その作品中の物体が，球や楕円体などの2次曲面を組み合わせたもので作ってあることが多いからである（もちろん反射・屈折しているからというのもあるが，反射・屈折はレイトレーシングでなくてもかなり優秀な近似表現をすることができるので，決め手にはならない）。メタボールの導入によって，扱える曲面は単純なものではなくなった。

メタボールはその性質上，単独で使ってもただの球(または楕円体)，あまりおいしくない（計算が重くなるだけ損ともいえる)。2つ以上を集め変形させることで初めてメタボールとしての特性を生かすことができるのだ。この一群のメタボールは，それ自身が1個のプリミティブのように働く。そして，ほかのプリミティブまたはメタボール群と論理演算ができる。形状ファイルを記述するうえでは，なんの不都合もなく通常のプリミティブとメタボールが共存することができるわけだ。

しかし，定義したメタボールが全部（それこそシーン中のすべてのメタボールが）くっつきあってしまったのでは，論理演算には使えない。論理演算をとりたいのに，その要素同士が融合・変形してしまっては困るのだ。メタボールのなかでも干渉する

ワイヤビュー

C-TRACEは昨年のバージョン3.0からモデラーがなくなった。テキストエディタでシーンを記述し，それを計算し，画像を生成する。どんな画像になるか，自分の思ったとおりの構図や色になっているか，そういったことはすべて計算が終わるまではわからない。いまさらいうまでもないことだが，レイトレーシングの描画は遅い。数時間から数十時間を要する計算なのに，できあがってのお楽しみ，というのは少々問題がある。とりわけ，メタボールは制御がかなり難しく，1回で狙ったとおりの形になることはまずない。

そこで，C-TRACE＋からは，本計算に先立ってできあがりの形を簡易表示してくれる「ワイヤビュー」がつくことになった。残念ながらまだ完成していないが，登録ユーザーには完成次第配布するとのこと。

形の制御は難しい

もの同士，干渉しないものを分けておくことが必要になる。

そこで，メタアトリビュートという新しい概念が導入されている。つまり同じグループに属するメタボールには同じメタアトリビュートをつけ，ほかと区別する。すると同じメタアトリビュートのメタボール同士のみが干渉しあう。あとは通常のプリミティブと同様に扱えばよい。ひとつのメタアトリビュートに属するメタボール群を，ひとつのプリミティブのように扱う。論理演算を行うことができるのは，そういうわけだ。

そして計算は重い◆◆◆◆◆◆◆◆◆◆◆

計算についても，これまでのプリミティブとは若干異なった処理が必要になる。先ほどもいったとおり，レイトレーシングでは，視線に最初にぶつかるプリミティブが見えることになっている。

視線は直線だから1次式。球などの2次曲面は，文字どおり2次式。したがって，球と視線との交点を求める場合は，2次方程式を解けばすむ。

ところが，メタボールの場合は，メタボールが作る濃度の分布を同じグループに入っているすべてのメタボールについて合計したものがしきい値になっている部分（曲面になる）と視線との交点を求めなくてはならない。言葉で書くとたいへん面倒そうだが，計算するのもたいへん面倒だ。プログラマが苦労するというだけでなく，コンピュータも苦労する。要するに，計算がかなり重い。2次方程式には解の公式があるが，メタボールの場合にはそれがないからだ。

まとめ◆◆◆◆◆◆◆◆◆◆

メタボールについてまとめておこう。

・メタボールはある点を中心として，中心部ほど濃く，周辺部ほど薄く分布する濃度球である。その濃度がある一定の値（しきい値）を超えた部分が見える。

・メタボールは，単独では単なる球だが，ほかのメタボールが近づくと互いに干渉しあい，変形する。

・負の重みを持つメタボールを作ることができる。ほかの見えているメタボールのそばに近づけると，相手のメタボールがへこんだり，穴が開いたりする。

・干渉しあうメタボールはひとつのグループとしてまとめる。このひと組のメタボール群は，ほかのプリミティブやほかのメタボール群との間で論理演算をとることができる。

＊

というわけで，メタボールの形を制御するためには，メタボールの原理を理解しておかなくてはならないし，メタボールの原理を理解するには，数学や物理をあまり嫌がらないことが必要なようだ（残念ながら）。

その他の拡張部分◆◆◆◆◆◆◆◆◆◆

メタボールという派手な部分の陰に隠れてしまってはいるが，C-TRACE＋にはほかにも拡張された部分がある。いずれもメタボールと直接の関係はないが，操作性や表現力を増すのに役立っている。

●スコープ機能

たとえば，画面の中で小さな物体が飛び回るアニメーションを作るとしよう。何十フレーム作るにしても，それぞれのフレーム中には背景などのようにまったく動かない部分があることが多いわけだ。したがって，これを2回以上計算（＝描画）することは明らかに時間の無駄である。アニメーションの全編にわたって，ほとんどの部分が同じ画像を何十コマも繰り返し計算することを避けることができれば，相当な時間の節約になる。そのために，必要な部分だけを計算させるのがスコープ機能。計算する前に，計算する範囲を指定することができるというものだ。

●スポット光源

いままでなかったのが不思議なくらいの光源。通常の点光源と違い，一定方向だけに照射する。文字どおりスポットライト。光の境界はぼかすことができる。

●αチャンネル

画像合成を美しく行うために，イメージファイル（画像ファイル）に新しく追加された情報。画像合成とは，ほかの画像を下敷きにし，その上にあらかじめ計算しておいたレイトレーシング画像を重ねて表示することを指している。これにより，たとえば複雑なシーンは各物体ごとに計算して最後に合成する，といったこともでき，計算時間の短縮を図ることも可能だ。

C-TRACEの出力するイメージファイル（～.IMG）には，画像の赤（R）・緑（G）・青（B）色の成分がそれぞれ記録されるのだが，今回新たにα成分が加わった。具体的には，αにはレイトレーシングで描画した物体の透明度や，アンチエリアシング処理で求めたピクセル占有率が格納されている。

プロテクトユニット

C-TRACE＋にはプロテクトがかかっている。といっても，通常のゲームソフトに見られるようなプロテクトではなく，かつてのZ'sSTAFFと同様，ハードウェア的なものである。システムユニットと呼ぶ部品をプリンタケーブルの途中にはさむことになっている。もちろんプリンタは正常に使うことができる。

Z'sSTAFFのプロテクトモジュールがジョイスティックポートを使っていたことからすると，多少はましといえようか。ちなみに，その他のグラフィックツールではどうかというと，Z'sTRIPHONYはマスターディスクをドライブ0に差してあるかどうかをチェックするようになっているし，外国ものの場合はマニュアルプロテクトという場合もある。

コンピュータグラフィックに限らないが，ツール類は，どうしてもハードディスクなどにインストールしたくなるものだし，バックアップをとることも必要だ。だからディスクにプロテクトをかけてコピーできなくするというわけにはいかない。

まあ，プリンタケーブルという普段触らない部分だから，一度つけてしまえばあとは気にする必要はない。

合成の際にこういう情報を持っておくと，ただ背景の上に重ねるだけでない高度な合成ができる。透明体は向こうが透けて見え，端の部分はアンチエリアシングがかかったまま重ね合わせられる。あたかも真面目にレイトレーシングですべて計算したように見える。むろん，ペイント系のグラフィックツールで描いたような絵の上に重ねても面白い。いろいろと応用がききそうだ。

右側がαチャンネルの内容

スポット光源の例

写真に合成した作例を載せておく。ただし，真面目に合成をしようと思ったら，計算の際には背景は真っ黒にしておくことが必要。イメージファイルに背景の色の成分が残っていると，合成が変になるからだ。今回のサンプルではそれをさぼっている。

問題は速度だ(TP版に期待)◆◆◆◆◆

この文章の冒頭で1カ月も試用期間があると書いたが，実際のところはほとんどものにならなかった。前衛芸術のような妙な物体ばかりごろごろとできてしまい，「自由曲面の造形」という目的からは遠いものになってしまった。ただ，どんな形ができるかわかりにくいというのは，メタボールの大きな特徴ではある。適当にいろいろな大きさ，いろいろなウエイトのメタボールを配置して，計算させる。変な形になることがほとんどだろうが，ときには意外に面白い形になることもあり，そこから新しい作品への発想も生まれるかもしれない。

メタボールは制御が難しく，できあがった形が変なものになる可能性は十分にある。一般論をいうと，メタボールで造形するには，局所的・近視眼的に形を作っても駄目。作りたい形状の本質を見抜いたうえで，できるだけ少ない数のメタボールを絶妙の位置に置いていくというのが理想。そこまで極めることができたら，メタボールを真に自由曲面として扱えるようになるだろう。

それにつけてもあの計算スピードはいただけない。結構解像度を粗くしたテスト計算でも，数時間から十数時間を要する。思いどおりの形になるまでは，メタボールの座標やウエイトなどのパラメータを少しずつ変えてやりなおさなくてはならない。たとえ数回しか計算しなおさなかったとしても，何日も要することになる。そのうち作品に妥協が生まれてこないとも限らない。

数値演算プロセッサボードはほしいところ。それほど極端に速くなるわけではないが，それでもあったほうがよい。なかったとしても，シャープ純正でないドライバ（FLOAT2.Xの改良版）のなかには速いものがあるので，それを手に入れるとよい（平気で2倍速かったりする）。当然，CONFIG.SYSなどの内容にも注意する。システムにはOPMドライバやBGプロセスなど遅くなりそうなものは組み込まない。

とはいうものの，いちばんいいのは早いところTP（トランスピュータ）版が出ることであるのはいうまでもない。なにしろ，計算が遅いというだけで，作品のクオリティが露骨に下がる（これは僕が悪い）。

C-TRACEは以前からトランスピュータバージョンも並行して発売している。トランスピュータは，とても速いCPUとでも思っていただければいいだろう。C-TRACE TPは，トランスピュータを積んだ基板と組になっている。そのトランスピュータボードを拡張スロットに差して使う。トランスピュータはレイトレーシングの計算だけをし，ほかの細かい処理（データのやりとりや画面表示）はX68000で行うことになる。

C-TRACE TPはC-TRACEとまったく同様に扱える。データのフォーマットは同じで，CPUが違っても互換性が保たれている。ただひとつの違いは，とにかく速いということ。ボクセル分割を導入したC-TRACEバージョン3（メタボールなしのC-TRACEのうちでは最も新しいバージョン）など，みるみるうちに完成するといってもいいくらいの速さであった。使ってみると，広告にある「170倍」という数字を十分に実感できる（それだけX68000が遅いということでもあるのだが)。これくらい速いと，いろいろな形を試したり色を変えてみたりといったことが実に気軽にできる。

さて，こうなると当然C-TRACE TP+の登場が待たれるところだが，現時点ではまだ移植がすんでいないらしい。というわけでこのレビューはX68000そのもの（浮動小数点ドライバまたはエミュレータであるFLOAT＊.X）で計算している。メタボールはただでさえ恐ろしく重い。そのうえ，前回のレビュー（C-TRACEバージョン3.0)はトランスピュータを使い放題に使ったので，今回はなおさら遅く感じた。いまはトランスピュータがX68000の300倍速いといっても僕は信じてしまうだろう（もちろんそんなことはない)。トランスピュータでメタボールを扱えるようになるまでは，メタボールには手を出さないのが賢明かもしれない。C-TRACE+はメタボール以外にも拡張されているのだ。

C-TRACEシリーズの名前がたくさん出てきたので，最後に整理しておく。

・C-TRACE 68 ver.3.0

C-TRACEの基本セット。主に2次曲面をプリミティブとして扱える。ボクセル分割を導入して高速化を図っている。

・C-TRACE TP ver.3.0

C-TRACE ver.3.0のトランスピュータバージョン。

・C-TRACE 68+ ver.1.0

メタボールやスポット光源などを導入して，C-TRACE ver.3.0を拡張したもの。上位コンパチブル。これを今回紹介した。

登場間近

CARD PRO-68K ver.2.0

FIXERの登場でX68000の環境にも変化の兆しが見えてきた今日この頃。うれしいことにCARD PRO-68Kのver.2.0が登場するというお知らせです。評価の高いデータベースのバージョンアップだけに期待を込めて速報をお届けしましょう。

Ogikubo Kei 荻窪 圭

なんだかんだいって，FIXER ver.4.0+microEMACSで原稿を書くようになってしまった。FIXERを使うと大人のためのX68000で扱うようなアプリケーションに対応していないものが多くて困る。WP.XやHyperwordはまだいい。原稿はエディタで書けばいい。厄介なのはKamikazeだ。

FIXERの状態でKamikazeを立ち上げるとしよう。標準ワープロのように日本語入力ができないが，そのときは実際それだけで済む。数字を打ち込むのに日本語FEPはいらないからだ。

問題はKamikazeを終了したあとである。なんと，変換モードに入れなくなってしまうのだ。立ち上げ直さないと駄目なのだ。これは困ったことである。アプリケーションの中でFIXERが使えないのは，Kamikazeのせいではないけれど（使えるようになっていたって罰はあたらないが），Kamikazeを抜けてまでFIXERが使えないのは，明らかにKamikazeのせいである。ちょっと困った。

さて，そういった緊迫した世界情勢の中で私は今月何を書くか。先月の予告もまたなんのためらいもなく破棄されてしまうのであった。

このコーナーで，X68000用アプリケーションの有意義な使い方やら実用情報やらを期待している人にはまことに申し訳ないのだが（私もそのつもりもあってこの連載を始めたのだし），今月もまた“活用法”はやんないのであった。ごめんなさい。

何をするかというと，ぎりぎりになって到着したCARD PRO-68K ver.2.0の制作途中版なのであった（以後，予告編と呼ぶ）。映画の予告編のような，「制作快調！」と字幕でも出てきそうな状態なのだが，なかなかver.1に比べて大いなる進歩が期待できそうなので，思わず紹介してしまおうというわけだ。

実際に製品版がくるのはしばらく先になりそうだが，もし届いたら，ちゃんと活用法付きでレビューしたいと考えている。

CARD PRO-68Kのバージョンアップ点

いきなりだが，計算やグラフや表入力はもはやスプレッドシートの特権ではなくなった。世間一般の常識として，表計算など数値を扱うものやグラフ化したいものはスプレッドシートで，大量の文字データや，定型印刷したいものはカード型データベースで，という風潮があった。しかし，それもまた崩れようとしている。スプレッドシートはデータベース機能を持ち，カード型データベースは計算機能を持った。

そして，CARD PRO-68Kはver.2.0になって，表形式の一覧表入力と，グラフ機能を持つにいたったのである。じゃーん。

だいたい，数値の計算でも，スプレッドシート向きのものとカード型データベース向きのものがある。伝票形式の元データがある場合，たとえば伝票や領収書や個別の成績カードなどは，元の形のままデータベースに入れておいたほうが，さまざまな用途に使えるのだ。もし，必要な計算やグラフ機能，印刷機能を備えていたなら，スプレッドシートよりカード型データベースのほうが向いているかもしれない。

さて，CARD PRO-68Kのバージョンアップ点で，ひと目でわかる大きな違いは，画面構成だ。なんとなくしょぼかった（失礼！）前バージョンに比べ，かなりかっこよくなった。色合いもクリーム色っぽいのから，グレーを基調とし，ボタンやメニューバーも立体的になり，ちょっとSX-WINDOWっぽいものになった。SX-WINDOWをどこかで意識しているのは確かで，ウィンドウのクローズボタンも，タイトルバー左の／ではなく，右の×にかわった。左にはズームアップボタンがある。

それだけでなく，前バージョンで見られた，消したり描いたりが目に見えるウィンドウ描画が，格段に速くなり，マルチウィンドウっぽいものになったのも見逃せない。概して，操作は軽快だ。

メニューバーからずりっと現れるプルダウンメニューも軽快。

メニューバーは固定で，個々の機能の操作は開いたウィンドウのタイトルバーの下にあるメニューからプルダウンする。一番上の段までマウスをぐりぐり運ぶ必要もなく，わかりやすくて便利だ。プルダウンメニューの現れ方も，トイレットペーパーを引き出すような（あ，ちょっとたとえが悪かったか），スリットから紙を引き出すような感じでいい気持ちだ。

でだ，こういった見た目の違い（詳しくは写真を見てちょ）に見合うほど，中身は変わったか。

1) グラフ機能

7種類のグラフ。立体グラフのないのがかなり残念だが，グラフ作成機能はKamikazeを超えるクオリティを実現している。種類は，折れ線，2次元分布，縦棒，円，レーダーチャート，折れ線&マーク，折れ線&縦棒とありがちなもの。いまひとつ種類が少ない気がするが，とにもかくにも，3次元立体グラフがほしいのは私のわがままだろうか。まあ，いいや。

グラフ描画はカラーでもできる。画面で見るときは，カラーのほうがいいね。グレー地にカラーってのはかなりシブい。

では，どこがKamikazeを超えているのか。それは，グラフ表示の柔軟性である。Kamikazeでグラフを描く際，グラフウィンドウの幅を狭くしたり，項目数が多いと横軸の項目表示が乱れるという指摘を以前，した。CARD PRO-68K ver.2.0では，文字フォントの大きさを変更することによって，グラフウィンドウが小さいときには小さい文字という当たり前ながらうれしいサポートをしてくれたのだ。

2) 一覧表画面での入力

CARD PRO-68Kの欠点として，一覧表画面は参照だけで，データ入力ができないという点も指摘されていた。ある程度のデータをまとめて入力する際，一覧表画面のほうがはるかに効率がいい。また，スプレ

ッドシート感覚で扱えるため，非常にとっつきやすい。

一番いいのが，同じ内容をいくつも入れたいとき，一覧表画面だとカット&ペーストで楽ができる（よね？　というのは，まだその機能は動いてなかったのだ)。

X68000というのは企業でおじさんたちがデータ分析や給与計算に使うマシンではなく，もっと小さな自営業者や，自宅にパソコンを持つ趣味の人がいろいろ工夫をしてパーソナルコンピューティングしていくためのパソコンである。これは断言する。だから，カード型データベースといえども，企業に置くと喜ばれそうな，使い勝手よりもデータの管理や保守を優先するソフトではなく，もっと気軽に使えるものでなくてはならない。

たとえば，データを追加しているときに，ふと前に入力したものも変えたくなったら，その変更したい項目をダブルクリックすると変更モードに自動的に入るとか，参照モードでも簡単な操作でデータ変更できるようにすべきなのだ。CARD PRO-68K ver.1では参照モードと追加モードと変更モードがはっきり分かれていた。今回のver.2では，ウィンドウの左上に現在のモードが表示され，プルダウンメニューで簡単にモードを切り替えられるようになったのはいいが，私が望むような簡単な入力がサポートされているかどうかは，今手元にある予告編ではわからない。

だいたい，データベースのコマンドを追加，削除，参照，変更とはっきり分けるのは大型コンピュータで，データベース操作言語を使ってデータベースを操作していた頃の名残にすぎないのではないだろうか，って最近は思う。レコードの削除だけは危ないので別にする必要があるが，そのほかはもう少し柔軟にすべきだろう。

あまりに簡単にデータを書き換えられると，人為的ミスが起きやすいと思ったなら，環境設定コマンドに「モードの自動変更有無」を設定できるようにすればいいのだ。

ちなみに，かなり毛色の違うソフトではあるが，MacintoshのHyperCardでは，ユーザーレベルをいくつか設定し，それによって書き換えられるレベルを管理している。

3）キーマクロ

これはうれしい。CARD PRO-68Kのプログラム機能が優れているといっても，プログラムを書くのはちょいと面倒だった。しかし，今度のはキーマクロがサポートされたのだ。使い方は簡単で，プルダウンメニューでキーマクロの記録開始を選んで，データベース操作をし，記録終了するだけ。

で，下手をするとこのテのキーマクロはひとつしか登録できなかったりするが，キーマクロのセーブや読み込みができるおかげで，非常に簡単にいくつものマクロを登録できるようになった。

4）プログラム機能

まだ全貌はわからないが，前バージョンで計算システムという名前がついていたプログラム機能も，いろいろ強化されたそうだ。予告編ではそのへんは動いていないようだった。

5）簡易ワープロ機能

ワープロなんていうから話がややこしくなるのであって，帳票作成と思えばいい。指定した点にデータベースのデータを貼り込み，あとは罫線を引いたり文章を入れたりする。

罫線はマウスでちょちょいと引ける。予告編では付属ワープロほど罫線の種類はなかったが，斜め罫線サポートという技がついていた。写真のように，斜め罫線が引けるのだ。また，4倍角がないのは悲しいが，1/4 倍角などもある。

ちょっとした表を打ち出したいとき，見積書のようなものとか，カセットレーベルとか，時間割りとかを作るときは，ワープロやKamikazeよりこちらのほうが便利かもしれない。というか，カード型データベースはそのくらい軽い使い方もできたほうがいいと思う。

6）データコンバート

前もあったけど，今回のほうがいろいろと指定が楽になった。区切りありASCII，区切りなしASCII，CSV(K3)，隼，1-2-3，SYLK，そして，CARD PRO-68K ver.1。

7）電卓

電卓はかっこよくなって，ルートや%もついた。普通の電卓なのだが，思わずそのまま同じものをSX-WINDOWにほしいくらい洒落ていると，付け加えておこう。

8）その他

予想されていたことだが，メインメモリ2Mバイト以上でないと，動作しない。

背景に絵を表示できる。できてなんになるんだ，という意見もあるが，できたっていいじゃないか。CARD PRO-68Kは当たり前だが768×512ドットのモードで動くので，65536色グラフィックの表示はできない（そこまでサポートはしてないだろう）と思う。どんなフォーマットのグラフィックをサポートしているのだろうか。

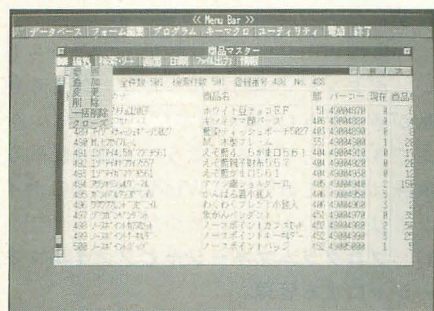
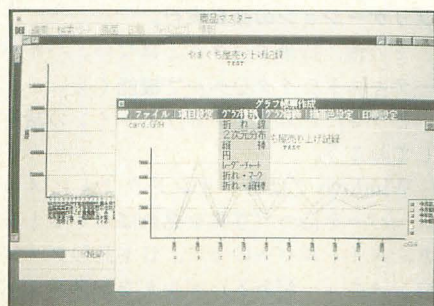
チャイルドプロセスがついた。

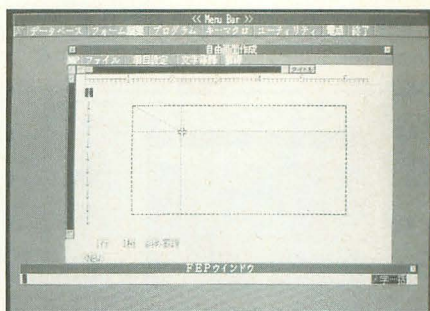
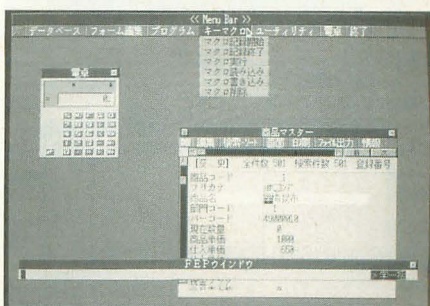
価格は未定。

FIXERには対応しているそうだ。FEPウィンドウが開いて，それがぐりぐり動くのに，大丈夫だろうか，て思ったが，予告編でもちゃんと動いていたので大丈夫のようだ。

*

個人的にはCARD PRO-68K ver.2.0が確定申告に間に合わなかったのは残念だ。今年もKamikazeでやるか。

もし，これで，データ入力時のレスポンスがよければ（前のは，大量データを打ち込むのがちょいと面倒だった)，使えそうだ。そうしたら，この連載で，ちゃんとレビューする。

一覧表入力が可能に

斜め罫線が使える

うれしいグラフ機能

フルタウンメニュー

SX-WINDOW vs Mac/Windows 3.0

WINDOWシステム大比較

Ogikubo Kei 荻窪 圭

ウィンドウシステムを持って生まれたMacintosh，数年間の苦行の末ついに満足のいくレベルにたどりついたWindows 3.0。これら2つのウィンドウシステムとの比較のうえでSX-WINDOWの行方を占ってみよう。

祝！　荻窪家Macintosh導入記念。といっても，過去のマシンとなってしまったMacintosh Plusだよん。ていうわけで，この原稿は，そういう事情を反映して書かれることになったのである。

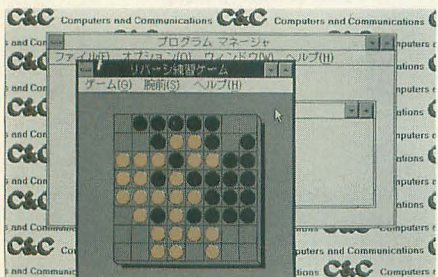
もう少し能書きを垂れるならば，他のシステムの様子をうかがうことによって，SX-WINDOWのよいところと悪いところ，次に進むべき方向でも見えてくれば幸いである。といったところか。

＊

X68000にとってみれば，ウィンドウシステムといえばもちろんSX-WINDOWであるが(KO-WINDOWというのもあるらしいが，それはこの際置いておく)，世間にはSX-WINDOWより有名なウィンドウシステムがたくさんある。

SX-WINDOWはそいつらに比べてどうなのか，どこが優れておりどこが劣っているのか。とまあ，ユーザーインタフェイスを中心に，そんな話をするわけだ。

対象とするのは（あまりたくさんあっても大変なので），1月23日に某ホテルの「平安の間」で大々的に発表された日本語Windows ver.3.0と，これ抜きでは語れないMacintosh＆漢字Talk，そしてSX-WINDOWである。ときどき，昨年秋に発表されたNeXTver.2.0の話も混じるが，さすがにNeXTを自由に使う環境にはなく，英語マニュアルは私の手に負えないので，詳しく触れたりはしない。

まずは，X68000ユーザーに贈る，各システム基礎知識から。

プログラムマネージャがWindows 3.0の基本

タスクスイッチャ

1. 日本語MS-Windows ver.3.0

パソコン誌では大騒ぎのWindows3.0である。先日，他社に先駆けて日電が「日本語Windows ver.3.0」を発表した。もちろんPC-98専用である。どうして大騒ぎかというと，Windowsしか大きな話題がないからである。それでもって，それまでのver.2.11がいくら騒ぎたくても，騒げるほど使いものにならなかったからである。ver.2.11がどのくらい使いものにならなかったかというと，「ファイルコピー時に，コピー先のディレクトリをキーボードから入力しなければならない」くらい使いものにならなかった。

今度のWindows3.0は随分とすごいらしい。ボタンが立体的になったり，使える色数が増えたり(16色が基本だが)，アイコンができたり，である。こういう週刊誌的な興味は置いておいて，Windows3.0のポイントはいまのところ，2つだ。

1）　プログラムマネージャとファイルマネージャによる管理。

2）　タスクスイッチャによる，MS-DOSアプリケーションの複数実行。

その他，特筆すべきことはない。ハイパーテキストライクなヘルプ機能くらいか。

どうせだからウィンドウデザインの話をしておこう。

Windowsでは各ウィンドウごとにメニューバーを持つ（写真を見ればわかるとおり，タイトルバーの下に，日本語のプルダウンメニューがある)。プルダウンメニューはドラッグしている間だけプルダウンするものではなく，一度クリックしたら開いたままになる。押したままドラッグする必要はない。

タイトルバーの左端にあるボタンはダブルクリックするとクローズ，シングルクリックだとプルダウンメニューが出る。ここにはウィンドウのアイコン化やズームアップ，移動などがある。なお，アイコン化(ミニマム）は「最小」，ズームアップ（マキシマム）は「最大」というように，“翻訳”されている。右端にもボタンがあり，こちらはマキシマムやミニマムの専用ボタンである。

話を戻す。1)のプログラムマネージャとファイルマネージャである。Windows3.0の第1のポイントだ。これはSX-WINDOWを動かしている数々のマネージャのような縁の下の力持ちと違って，しゃしゃり出てくる，目立ちたがりやのタスクだ。

プログラムマネージャというのは，プログラムの管理をし，ファイルマネージャというのはファイルの管理をする。どうしてこれがポイントかというと，ほかのパソコン用ウィンドウシステムでは両者がここまで分離してないからである。

Windowsの基本はプログラムマネージャである。プログラムマネージャウィンドウがあり，こいつをクローズすることはWindowsの終了にほかならない。

このウィンドウの中にはさらに小さなウィンドウやプログラムアイコンがつまっており，そのアイコンをダブルクリックするとプログラムが起動する。だが，プログラムマネージャからプログラムを起動させるには，プログラムマネージャにプログラムを登録せねばならない。また，プログラムマネージャはプログラムの実行をマネージメントするだけであるから，ファイルの操作はまったくできない。このあたりがわかりにくいところだ。

ここで，ファイルマネージャが登場する。ファイルマネージャはWindows2.11みたいである。ウィンドウの中にドライブアイ

コンが並び，指定したドライブの中のディレクトリがツリー上に表示される。一度に見られるドライブはひとつだけである。ただし，ツリー状に表示されたディレクトリをダブルクリックするとディレクトリウィンドウが開く。このウィンドウはいくつも開く。ファイルコピーするときは，コピー元のディレクトリウィンドウを開き，コピー先のドライブを指定して，ファイルをドラッグする。

両者が分かれていると，非常に面倒くさい。プログラムマネージャにプログラムを登録するときは，アイコンを指定し，アイコンの名前を指定し，それをダブルクリックしたときに起動するファイルのありかを指定する。これで大丈夫だ。

ここで，ファイルマネージャを開き，さっきプログラムマネージャで指定したプログラムの場所を，別のドライブへ移したとしよう。

そうすると，プログラムマネージャとファイルマネージャは連動していないので，プログラムマネージャ上でダブルクリックしても，起動してくれないのである。

変なの。

でも，プログラムマネージャにもいいところはある。ひとつは，SX-WINDOWやMacのようなアイコン表示ができること。ファイルマネージャではできない。

もうひとつは，アイコン名の制限がとってもゆるいこと。MS-DOSのように8文字＋拡張子3文字という制限はなく，長い名前をつけられるので，便利。

アプリケーションを起動するだけのエンドユーザーのための仕様だ。

ウィンドウシステムのメリットのひとつが非常に“ファイル管理が楽になること”と考えていた私にはショックではあった。

さて，2)のMS-DOSアプリケーションの複数実行というのも重要である。タスクマネージャのおかげで，8086用に書かれたアプリケーションをいくつも実行できるのだ。そして，386エンハンストモードってのを使うと，マルチタスクまでしてくれる。

プログラムマネージャからDOSのプログラム（一太郎とか1-2-3とか）を実行すると，それはWindows3.0のタスクがひとつ増えたのと同じ状況となり，一太郎を終了させなくてもタスクスイッチを押すと，プログラムマネージャの画面に戻る。そして，1-2-3を立ち上げたとすると，裏で一太郎も立ち上がったままで，1-2-3も使えるのである。

これは便利である。メモリはたくさん食うけど。そうでなければハードディスクとの間で頻繁にスワッピングすることになって，遅い。

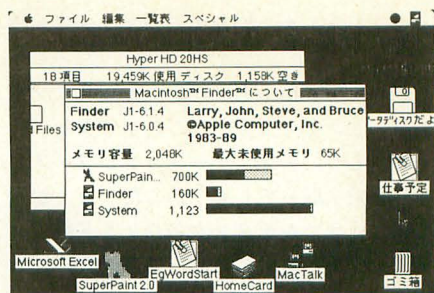
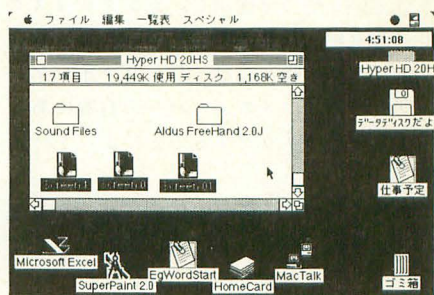
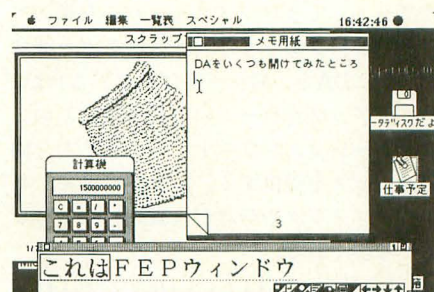
とまあ，いろいろWindowsの話をしたが，ポイントはまだあって，本家IBM PCのWindowsは640×480ドットが事実上標準の解像度となっているのに，PC-9801は640×400ドットなため，“アプリケーションによっては困る”こととか，日本語が16ドット表示で行間が妙に空いているものだから，見た目のバランスがよくないといった欠点がある。

2. Macintosh 漢字talk

いわずとしれたMacintoshである。WindowsがDOSのさまざまな環境を引きずらねばならなかったのに対し，Macintoshは最初からMacintoshであった。とはいえ，Macintoshのシステムも度重なるバージョンアップの末，かなり大きなものになっている。Macintoshで日本語を使うためのシステムのひとつ，漢字Talkを使うとしたら，メインメモリが2Mバイト以上必要で，これはSX-WINDOWと同じ。

周知のとおり，Macintoshはワンボタンマウスであるから，ポップアップメニューという芸当ははなからできない。メニューはすべて画面最上段にあるメニューバーからプルダウンすることによって選ばれる。カット&ペーストという基本的な作業でさえマウスをずりずりと最上段まで持っていく必要があり（大きなディスプレイだと，メニューバーが遠くて疲れる），それを避けるためか，ショートカットキーと称してキーボードにもいくつかの機能が割り当てられている。

Macintoshのウィンドウシステムはファインダーと呼ばれるワークスペースを基礎としている。ファイル操作やアイコンの実行を行うのがファインダーだ。ファインダーはメニューバーとデスクトップより成っている。デスクトップというのは背景だと思っていい。デスクトップ上にはウィンドウやドライブアイコンのほか，ファイルのアイコンも置くことができる。この，ファイルのアイコンを置けるというのがひとつの操作上でのポイントとなろう。

実行したいプログラムというのは，階層ディレクトリ（Macintoshではフォルダという）の奥へ分散しているのが普通だ。そうなったとき，目的のアプリケーションが現れるまでディレクトリを探索し，ダブルクリック，っていうのは面倒である。そこで頻繁に使用するアプリケーションは，そのアイコンをドラッグしてデスクトップ上へ置いておけばいい。こうすれば，デスクトップ上のアイコンをダブルクリックするだけで，そいつを実行することができる。ディレクトリのウィンドウからデスクトップへアイコンを移動すると，元のウィンドウからアイコンが消えるので心配になるが，ファイルが削除されたわけではない。楽屋で休憩からステージ横へ待機に変わるようなものだ。

Super PaintとFinderを開いただけで，残メモリ65Kバイトというメモリ食いのマック。

デスクトップにおいたアイコンと右上のファインダアイコン（マルチファインダ時のみ）。

メニューバーの時計はシェアウェア。フォントを選べるFEP。グラフィックも入るスクラップブックなどもある。

Macintoshはこの機能によってMS-DOSやHuman68kのコマンドシェルでいうパスを不必要にし，Windowsのようにプログラムマネージャを別に用意するという技に頼らなくてもよくなっている。偉い。さすがだ。

問題はSX-WINDOWにこの機能に相当するものがないことだ。X68000アイコンというものはあるが，あれはひとつのディレクトリにすぎない。アプリケーションのア

イコンをまとめて管理するウィンドウが欲しいと思う。

さて，Macintoshは一度に複数のアプリケーションを立ち上げることができない。1つひとつのアプリケーションがファインダーを乗っ取ってしまうからだ。ここが一般的にいうマルチタスクのウィンドウシステムとは異なるところだ。基本的に，シングルタスクのシステムにすぎない。

が，最近のシステムではマルチファインダーというやつを採用し，シングルタスクではあるが，複数のアプリケーションを立ち上げられるようになった。ファインダーの切り替えは，メニューバーの右端にあるアイコンをクリックすることによって行われる。

マルチファインダーの欠点は，メモリを食うこと。2Mバイトでは使いものにならない。

Macintoshはマルチウィンドウでない部分を，デスクアクセサリというシステムでサポートしてきた。メニューバー左端のアップルマークがそうだ。デスクアクセサリ（一般にDAと，略称で呼ばれている）はいつでもアップルマークの下にあり，どのアプリケーションが立ち上がっていても，同じサービスを提供する。計算機やメモ，スクラップブックや時計が代表的。シェアウエアも多くある。

DAのほか，INITとかCDEVと呼ばれるプログラムもある。INITやCDEVは起動時に常駐し，いろんなことをしてくれる。漢字Talkもその一種。

Macintoshで秀逸なのは，INITもCDEVも，システムフォルダにコピーするだけで次の起動からきちんと働いてくれるし，DAもFONT/DAムーバーというアプリケーションを使ってシステムに登録すればいいだけということだ。インストールが驚くほど簡単なのだ。日本語FEPもプログラム本体と辞書をシステムフォルダに入れて，コントロールパネルというDAを開いて漢字Talkアイコンをクリックし選択すればいいだけ。DOSやHuman68kのような面倒なことはない。

欠点といえば，面倒なことをしなくてもいい代わりに，DOSやHuman68kではユーザーがすべきかなりの部分をシステムやプログラムに任せてしまうこと。ウイルスに狙われやすいとか，「MacintoshはMacであって，パソコンではない」となってしまいがちだ。徹底的に道具としてのパソコンを追求しているため，かえって気持ちがいいという面もある。

3．SX-WINDOWとの比較

以上がそれぞれのユーザーインタフェイスを中心とした概要である。SX-WINDOWについては言わずもがな，ということで割愛させていただいた。

次に，この3つを比較してやろうと思う。

●ファイル管理

SX-WINDOWのファイル管理はMacintoshを徹底的に参考にしている。複数ファイル選択の方法（ドラッグして囲むとか，SHIFTキーで追加するとか）など，Macintoshそのものだ。同一ドライブなら移動で，異なるドライブならコピーというのも一緒。ファイル削除はゴミ箱（SX-WINDOWはクリーナー）というのも一緒だ。ちなみにNeXT ver.2.0がしょってる。ver.1ではブラックホールだったが，ver.2ではリサイクラになったのだ。リサイクルってセンスがさすが。

MS-Windowsは少々毛色が異なる。ウィンドウシステム自体がファイル管理をするのではなく，ファイルマネージャというウィンドウ上で走るプログラムがファイル管理をする。だから，背景にはゴミ箱もなければドライブアイコンもない。ただ絵があるだけ。ドライブアイコンは，ディスクの有無にかかわらず，接続されている全ドライブのアイコンがファイルマネージャの中に並び，クリックしたドライブの中身（サブディレクトリ）がツリー状に並ぶ。そんでもって，ツリーの中のサブディレクトリをダブルクリックすると，やっとディレクトリウィンドウが開く。このウィンドウはファイルマネージャの管理下にあるため，ファイルマネージャウィンドウの外に出ることはできない。ディレクトリウィンドウはいくつも開くので，2つ開けばその間でファイルのコピーなどができる。はあ，言葉で説明すると，なんて複雑。

NeXT ver.2.0のファイル操作なんてお洒落なのだが，説明が面倒なので割愛。

●ドライブのイジェクト

Windowsに関しては，相手がDOSマシンであるからして，ディスクのイジェクトなんてできるわけがない。

Macintoshは一応，プルダウンメニューからイジェクトを選んだり，アップルキー＋Eでイジェクトできるのだが，Macintoshユーザーはそんなことはしない。ドライブアイコンをぐりぐりとドラッグして，ゴミ箱に放り込むのである。すると，イジェクトされる。ゴミ箱に捨てると吐き出される，ってことは，あのゴミ箱は現実の世界とつながっているのだろう。

NeXTもver.2.0からその技を踏襲している。ドライブのアイコンをリサイクラに放り込むと，イジェクトだ。

SX-WINDOWはもう少し凝っていて，ドライブアイコン自体にイジェクトボタンがついている。これを押す。マウスの移動量が少ないという点で，SX-WINDOWが一番である。

●プログラムの実行

GUIってやつに人々が一番期待するのは，ファイル管理と，アプリケーション管理である。Windowsは見事にその2つを分けてみせたし，Macintoshは見事に融合してみせた。SX-WINDOWはいささかこの点において情けない。

DOSやHuman68kというのは，実行ファイルの位置を探さなくてもいいように，パスというものとバッチファイルというものを考え出した。エイリアスってのもHuman68k 2.0では使えるようになった。しかし，これは，コマンド名を入力することのないウィンドウシステムとは相容れないものである。

じゃあどうすればいいかというと，アプリケーションの実行を指示するアイコンが，いつでも手の届くところに，必要なアプリケーションの数だけあればいいのである（エイリアスアイコン！ってのはどうだ）。

それがMacintoshでいうデスクトップにファイルのアイコンを持ってきてしまうことであり（デスクトップだけはファイルのウィンドウが全部閉じても残っている），Windows3.0がとった，プログラムマネージャというプログラム実行専用のマネージャを作って，そこに登録してしまえ，っていう方法である。

じゃあ，SX-WINDOWはどうか。今のところ，そういう工夫は見られない。なんとかしてドライブトレイみたいな，アプリケーショントレイ（アプリケーションを実行するだけのアイコンがある）でも作れないかしら。

あと，Windows3.0には実行中のウィンドウのアイコン化って技がある。使っているプログラムがちょいと画面上で邪魔になったけれどクローズはしたくないとき，ウィンドウのアイコン化って技を使うと，実行中のウィンドウがアイコンになって，左下に収まる。これは，NeXTでもできる技だ。マルチタスクシステムなら，このくらいは欲しい。Macintoshはマルチファインダーって技を使うので，この機能はない。

●システムを動かすのに最低限必要な環境

一番安いのはなんといってもMacintoshである。そういう時代になってしまった。Macintosh Classicっていう最廉価のマシンでもハードディスクさえつなげばちゃんと動く。純正ハードディスクを内蔵したMacintosh Classicは定価が298,000円だ。ディスプレイ，キーボード，システム込み。

続いて，X68000かな。ディスプレイ＋本体＋2MバイトのRAM＋ハードディスクとしよう。PROII HD＋1Mバイトの増設RAM＋一番安いディスプレイで，定価507,800円だ。

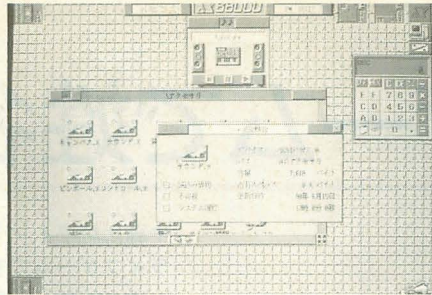
Windows 3.0へと移ろう。こいつの場合はいろいろランクがあるが，満足に使うための最低限のシステムとして，32ビットのCPU，2MバイトのRAM(本当は4Mバイトといいたいところだけど，妥協して)，ハードディスクだ。それで，カラーディスプレイ。概算すると，エプソンのPC-386MがHDモデルで468,000円。でもエプソンの機械で日電のWindowsは動かない。日電純正なら，新製品のPC-9801DS (CPUは386SXの16MHzという，32ビットでは一番安いやつ）の40MバイトHDDモデルが508,000円。増設RAMはよくわからないが，2Mバイトで30,000円くらいだろう。ディスプレイがまあ，X68000と同じくらいとして，約90,000円。計628,000円となる。

MacintoshとWindows 3.0については，贅沢をいいだせばきりがないが，X68000の場合，きりがある。うーん。

●マルチメディア度

音と映像。こういったものが扱えればマルチメディアと称するらしい。どの程度扱えれば認められるのかというと，まあ，音はPCMかAD PCMくらい必要だろう。映像も，最低256色から数万色が必要だろう。

Macintoshはカラーの機械を使えば，機種による差はあるが，640×480ドットで256色というのが一般的だ。もっとも，セットするビデオボードには640×870ドットや，24ビットカラー（1600万色）などどうにでもなる。ちなみに，IBM PCもビデオボードを入れ替えればどうにでもなる設計になっており，今はVGAグラフィック(テラも採用した，640×480ドットで16色。あるいは320×200ドットで256色)が一般的であるが，スーパーVGAやXGAなどもっと細かいのもある。どうにでもなる設計って，いい。

ウィンドウシステムということで見ると，SX-WINDOWで扱えるのは640×480の16色。IBM PCのWindows 3.0レベルにとどまっている。ただこれはキャンバス.Xの話で，768×512ドットの16色のほか，65536色だって可能だ。

ファイル情報

日本語Windows 3.0は日電の場合，640×400ドットの16色か，ハイレゾモード（H98ってやつ）になる。

音っていえば，X68000はFM音源＋AD PCM，MacintoshはPCM音源を持っているので，問題はない。下手をするとビープ音しか鳴らないPC-9801は問題外。

●アプリケーション

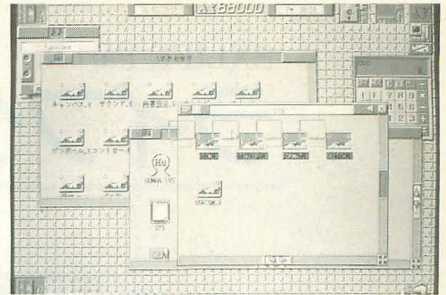
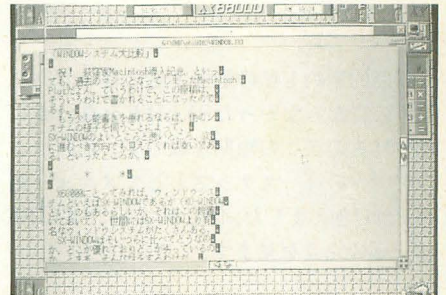
SX-WINDOWについてはいわずもがな。ただし，従来のアプリケーションもダブルクリックでそのまま実行できる。メインメモリが2Mバイトだとメモリ不足のケースも多くなるだろう。4Mバイトは欲しい。

こういった状況はWindows 3.0でも同じ。従来のアプリケーションも実行できる。ただし，Windowsの場合，PIFファイルというDOSアプリケーション実行用の設定ファイルが必要である。これがけっこう面倒だったりする。しかし，メモリが4Mバイトほどもあればタスクマネージャのおかげでいくつものアプリケーションを実行し，切り替えて使うことができる。コマンドシェルウィンドウもある。

Macintoshの場合，過去の資産のしがらみがないので，問題はない。

ウィンドウシステム用アプリケーションといえば，Macintoshが一番多いのは当たり前。次がWindowsだが，これもそうたいしたものはまだない。タスクスイッチャを使ってWindowsから立ち上げる一太郎というのがとりあえずの使い方だろう。

4. 私観

Macintoshユーザーは数多いが，Macintoshが好きでMacintoshユーザーになったMacintoshユーザーのMacintoshを見ると，誰ひとりとしてデフォルトのシステムでは使っていない。メニューバーの上に時計を常時表示したり，表示フォントを小さくしたり，背景にグラフィックを入れたり，スクロールバーの形を変えたり，終了するたびにスタートレックのオープニングが流れたり，ウイルスガードを入れたり。そういうものである。そういうシェアウェアも多く出回っている。

ファイルアイコンのドラッグ

ノート。Xを開いたところ

というわけで，SX-WINDOWに欠点があると思うなら，直してしまえばいいのだ。新感覚のスクロールバーが気に入らなければ変えてしまったり，時計を別のものにしたり，ファイル検索プログラムをアクセサリに追加したりすればいいのである。

WindowsはDOSマシンがそうであったように，企業ユースをかなり意識している。だから，MacintoshやSX-WINDOWのように柔らかくはなく，ファイル管理とプログラム管理を別にして，普段の使用では誤ってファイルを消してしまったりしないようになっている。あまり魅力を感じないのはそのせいだろう。ただ，日本語になるとどうしてああも画面のバランスが悪くなるのかは理解に苦しむところだ。

ウィンドウシステムにはほかにも，UNIXのだとか，OS/9のだとかあるけど，エンドユーザーが使うことについてどう考えているかが異なる。当初，エンドユーザーのことしか考えてなかったのがMacintoshであり，そのエンドユーザーを考えてないのがUNIX系のウィンドウシステム（NeXTを除く）やOS/9だったのだと思う。Windowsはどっちも睨んでいる。SX-WINDOWはどうだろうか。

というわけで，SXエンターテイメントキット計画をよろしく。X68000にあっているのはやはり，広い意味でのエンターテイメント・ウィンドウシステムだと思うのだ。

SXLIFE Part III

ライフゲームで姓名判断?

Nakamori Akira 中森 章

中森氏がウィンドウプログラミングの第一弾として取り組んだライフゲームですが、今回で完成となります。いつかの不都合な点を改善し，最後にライフゲームそのものでいろいろと遊べるようにしてみました。

時間的制約と私の勉強不足のせいで完全なものにならなかったSXLIFEを真の姿に近づけるのがこの短期連載です。前回はこの野望への第一ステップとしてメニュー処理を付け加えました,今回でついにSXLIFEは一応の完成を見ます。

バグ繕いなどのバージョンアップ

SXLIFEを作ってはみたものの，前回までに紹介してきたSXLIFEには，周囲からの要求や改良したい点，あるいは不具合が結構残ってしまいました。まずはこれらを解決してすっきりとしましょう。

●グローボックスの色がおかしい

SXLIFEはドットを色付きで表示することを目的としていましたから，アクセスページを4ページ全部使っていました。ただし，ウィンドウのグローボックスの表示はアクセスページ数を2と仮定しているみたいなので，SXLIFEのグローボックスはおかしな色をしていました。標準的な黒と灰色でグローボックスを表示するために，グローボックスをアップデートする前にアクセスページ数を2に戻してやることにします。具体的にはリスト1のようなサブルーチンでグローボックスの書き換えを行うように改造します。

●起動直後のハイライト表示

SX-WINDOWを立ち上げた直後，SXLIFEがアクティブになっていない状態では，SXLIFEのウィンドウをいくらマウスで選択してもハイライト表示されません(つまり，ウィンドウがアクティブになってもそれが目に見えない)。ただし，ウィンドウを一度移動すればちゃんとハイライト表示されるようになります。これはウィンドウのオープン後，自分のアクティブフラグ(SXLIFEのリストではactive(a5)という変数)を初期化していないために生じる現象です。

プログラムの記述上，アクティブフラグがメモリのゴミなどによって0以外(アクティブであるというしるし)になっていれば，いくらアクティベート要求が来ても自分を選択するための，

　__WMSelect

というシステムコールが発行されないためと考えられます。

対策としてはウィンドウのオープン後にアクティブフラグを0にしておきます。具体的には，

```
move.w #0,active(a5)
```

を付け加えるだけです。

●全クローズ要求が無視される

ウィンドウの全クローズ要求はシステムイベントとしてそれぞれのウィンドウに通知されてきます。SXLIFEではシステムイベント(タスクマネージャからのイベント，イベント番号12と13)の処理をまったく記述していないので全クローズ要求は無視されてしまいます。そこで，システムイベントの処理を既存のプログラム(謹賀新年PRO-68KのTMSAMP.S)を真似てリスト2のように記述しておきます。リスト2では全クローズ以外の処理(何かよくわからん)も記述してありますが，オリジナルから特に変える必要もないのでそのままにしてあります。

●「SXLIFEについて」のダイアログ

メニューの「SXLIFEについて」ではエラーメッセージ用のダイアログを使用していました。が，通常の説明にエラーメッセージ用のダイアログを使用するのは気持ち悪いので，べつのダイアログを使用します。PDSなどを眺めていたら説明用のダイアログとしてエラーメッセージ用以外を使用していたものがあったのでそれを利用してみます。システムコール，

　__DMRefer

の引数のDLOGのIDとして$F000を指定する(これは負の数だから本当は使用が禁止されている)とシステム専用の無地のダイアログがオープンできるようです。このウィンドウに，

　__GMShadowStrZ

というシステムコールで文字を書いていけば素晴らしい説明用ダイアログの出来上がりです。ダイアログが表示できたら，

　__DMControl

というシステムコールでボタンが押されるのを待つだけです。このダイアログ表示用のプログラムはリスト3です。ほかのプログラムでも流用できるようにここではSXLIFEのソースとは別ファイルにし，SXLIFEのソースからinclude疑似命令で取り込むようにしています。

このダイアログはかっこいいのですが，禁止事項を無視しているのがいけないのか，ダイアログを表示したあとはせっかく直したグローボックスの色が再びおかしくなってしまいます。いろいろ試してみた結果ダイアログをオープンする(__DMRefer)前にアクセスページ数を2に戻しておけばうまくいくようです(はっきりいってオマジナイ)。いずれにしても，使用禁止のIDですから今後はなにか支障が出るかもしれません。とりあえず作法が確立されるまでの手段と心得ておいてください。

●デモパターンが少ない(ない)こと

前のSXLIFEにはデモパターンがありません(立ち上げ時に既定のパターンが表示されるだけ)。前回でせっかく(ポップアップ)メニューを追加したのですから，メニューでデモパターンを表示できるようにします。リスト4がデモパターンを表示するプログラムです。現在，16種類のデモパターンを表示することができるようになっています。

レジスタd0にデモパターンの番号(メニューの選択結果からこの番号を作るようにする)を入れてdemoLifeというサブルーチンを呼び出すことで指定したデモを開始します。サブルーチンdemoLifeで表示するデモパターンは，表示位置，表示位置からのオフセット(相対位置)としてのドットの

座標，終了コードから構成されていて理論的にはデモパターンの数をいくつでも追加できるようになっています。

本当は表示開始位置はマウスカーソルで表示するようにしたかったのですが，私の限界を超えているようなので見送りました（誰か改造して）。その代わり，というわけではありませんが，SXLIFEの起動時にこのデモパターンを適当に選んで表示するようにSXLIFEの初期化部分を改造しました（これまでの既定パターンはやめ）。リスト5がその初期化の部分です。IOCSコールで得た時刻からデモパターンを決定するようにしただけなのですが，ウィンドウをオープンするたびに異なるパターンが現れてなかなか楽しくなりました。

●ドットの設定にマウスの右ボタンを使っていること

これは，どうすべきかちょっと迷ったのですが，日和見的な私は結局改造することにしてしまいました。SXLIFEはドットパターンの設定を右ボタンで行うようにしておきました。これは左ボタンの負荷を軽くするためと制御ボタンやメニュー項目の選択などによってモード切り替えを行う煩わしさを排除したためだったのです。しかし，編集部内ではこれが不評だったみたいで，マウスボタンの使い方がめちゃくちゃという意見もあったようです。

そこでほかのアプリケーションとマウスボタンの操作を統一することにします。すなわち，右ボタンはメニュー専用で，左ボタンはその他もろもろの指示や設定に使います。この改造自体はそれほど難しくはありません。ソースファイルの中から変更部分を適当に抜き出してくっつけたのがリスト6です。それでも，すんなりと改造するのは嫌だったのでリスト6ではささやかな抵抗を試みています。つまり，

RightSet

というシンボルで新仕様と旧仕様を切り分けることができるようにしてあります。このシンボルが定義されていれば右ボタンでドット設定をするSXLIFEになります。

それはともかく，左ボタンでドットを定義する場合は，設定を終了する方法を考えなければなりません。旧仕様では左ボタンで設定を終了という非常にシンプルな構造だったのですが，新仕様では左ボタンでドットの設定を行いますから別の方法が必要です。残る方法はウィンドウ内の制御ボタンかメニュー選択によるものですが，実現の容易さを考慮してメニュー選択を選択することにします。このとき，メニューに新たなメニュー項目を付け加えることはやめ，ドットを設定している間はメニュー項目の「ドット設定を開始」の部分の表示が「ドット設定を終了」に変わるようにしてみました。もちろんドットの設定が終われば，その項目は「ドット設定を開始」に戻ります。

これが100%のSXLIFEだ

さて，前置きがかなり長くなりましたが，問題点を処理して気分がすっきりとしたところで先に進むことにしましょう。それでは，私のイメージしていた100%のSXLIFEを紹介することにしましょう。それは，人の名前を初期パターンとしてライフゲームを行うことです（© 加藤賢哉）。

パソコンの文字フォントはいくつかのドットの集まりとしてROMの中に格納されています。人の名前は文字で表せます（当たり前だ）から，その文字フォントをROMから読み出してライフゲームの初期パターンとすることも可能なわけです。自分の名前のドットパターンを初期値としてライフゲームを行ってみて，それがあるパターンに収束するのか，あるいはすべて消滅してしまうのかを眺めるのは結構興味深いことではないでしょうか（占いになったりして）。

さて，名前のパターンでライフゲームを行うことになると必要になるのは名前を入力するためのウィンドウです。キーボードから文字を取り込むことができれば，それに応じたフォントをROMから読み出してウィンドウ上に表示するのは簡単にできそうです。そのため，名前を入力するウィンドウさえ作ってしまえば100%のSXLIFEは完成したも同然なのです。以下では，

- **名前を入力するウィンドウの作り方**
- **フォントパターンの画面への配置**

を順番に説明していきます。

名前を入力するウィンドウ

SX-WINDOWなどのようなウィンドウシステムではユーザーと情報のやりとりをするためにダイアログという機能が用意されています。したがって，ダイアログを使えば名前（一般的には文字列）の取り込みはいとも簡単にできるはずなのです。

しかし，私にはまだダイアログの使い方がわかりません。そもそも，謹賀新年PRO-68Kに付属してきたSX-WINDOWのドキュメントを完全に理解できた人は何人いるのでしょう。はっきり言って，私もまだドキュメントの半分も理解していないのです（難解というかよくわからない）。プログラム例さえあればドキュメントが難解でもなんとか理解できるものです。ところが，現時点のSX-WINDOW上のソフトではダイアログ機能をまともに使用しているものはひとつもありません。シャープから供給されているサンプルでさえ，ダイアログ的な処理が必要な場合は，ダイアログ用ウィンドウを使わず通常のウィンドウを使用しています。

そこで，私も今回はダイアログを使用するのはあきらめて，通常のウィンドウを使用して名前入力用ウィンドウを作ることにしました。そうと決まれば話は簡単です。名前を入力するだけなら，身近にTMSAMP（これは入力したコマンドを実行するサンプル）というプログラムがあるではありませんか（謹賀新年PRO-68Kに入ってたやつ）。これを解析すればなんとかなりそうです。

名前を入力するウィンドウといっても，所詮はSX-WINDOW上のウィンドウのひとつですからプログラムの基本はこれまでに学んできたものと変わらないはずです。案の定，ウィンドウの初期化に関してはウィンドウをオープンしたあとにテキストエディット用のメモリを確保する程度の違いしかありません。これはメニューを使用するときにメモリを確保したのと同じ考え方ですね。テキストエディット用のメモリ，

__TMNew2

というシステムコールで確保します。そして，このあとにイベント処理用のループに入ります。

イベント処理用のループで行うことは通常のウィンドウとほぼ同様です。ただ，テキスト処理用に，

__TMEvent

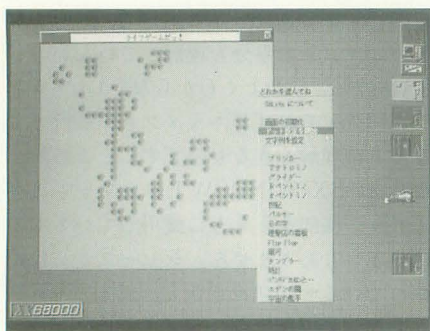
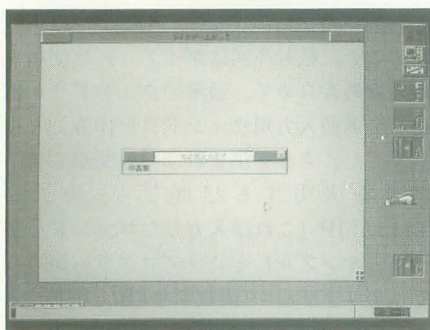
というシステムコールが何カ所かで使われています。これは，

- **ヌルイベントでのキャレット（カーソル）の点滅**
- **マウス左ボタンダウンイベントでの範囲選択**
- **マウス右ボタンダウンイベントでのカット・アンド・ペースト処理**
- **キーダウンイベントでの文字入力**

の4種のイベントを処理するためのシステムコールです。

したがって，これら4つのイベント発生時には基本的にはこのシステムコールを発行しておけばよいのです。ただし，マウス

メニューを開いたところ

どれかのデモパターンの画面

名前を入力するウィンドウを開いたところ

名前を入力して動き始めた直後

左ボタンダウン時はウィンドウの移動などの処理も行わなければならないため，通常の場合と同じく，

__SXCallWindM

を発行したあとで，

__TMEvent

を発行することになります。

この場合でもマウスカーソルがエディット範囲に入ってなければ意味がないので，

__EMMSLoc

でマウスカーソルのローカル座標を得たあと，

__GMPtInRect

によってその座標がエディット範囲内かどうかを調べ，範囲内であるときのみ，

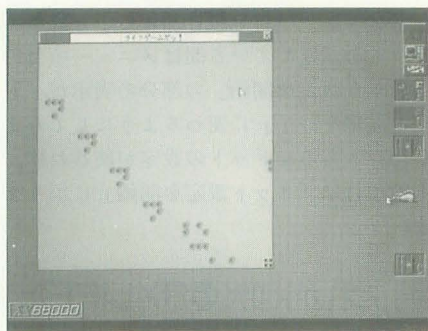
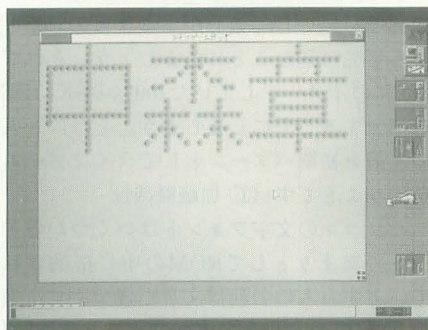
__TMEvent

を発行するようにしなければなりません。

また，キーダウンイベントではリターンコードが入力されたらウィンドウをクローズして入力された文字列を親であるSXLIFEのウィンドウに引き渡す処理を書いておかなければなりません。イベントレコードのwithフィールドの下位ワード（イベントレコード先頭からオフセット4のアドレス）が入力された文字のASCIIコードですから，それを$0Dと比較すればリターンコードの入力を知ることができます。

入力された文字列を取り出すためのシステムコールは，

__TMGetText

です。このシステムコールの返り値によって実際に入力された文字数（バイト数）を知ることもできます。文字列を親ウィンドウに引き渡したあとは，自分自身をクローズしなければなりません。この場合，

__WMClose

または，

__WMDispose

でウィンドウをクローズする（使い分けはわかりますね）ほかに，

__TMDispose

によって確保したテキストエディット用のメモリを解放しなければなりません。

上の4つのイベント以外でどうしても処理をしなければならないのはアクティベートイベントとアップデートイベントです。それでは，残りのイベントの処理を見ていきましょう。

●アクティベートイベント

アクティブフラグをセットするだけです。これは通常のウィンドウとまったく同じです。

●アップデートイベント

テキストエディット用の表示領域を書き直すシステムコールは，

__TMUpDate

ですから，アップデートイベントの処理では，このシステムコールを定型的な，

__WMUpdate

と，

__WMUpdtOver

のシステムコールの間で発行すればよいのです。

ただし，これらのシステムコールを発行する前に，

__TMCaret

によってキャレット（カーソル）を消しておかなければなりません。さもないとキャレットの跡が画面の至るところに残って悲惨なことになるでしょう（特にウィンドウを移動する場合）。

また，この場合は子のウィンドウ（名前の入力）のアップデートですが，子のウィンドウがオープンしている間に親ウィンドウ(SXLIFE)のアップデート要求が発生した場合は親をアップデートすることを忘れてはなりません。親のアップデートを忘れると子のウィンドウを移動したとき，その子ウィンドウが元あった位置が（親と重なっていたなら）すっぽりと歯抜けの状態になってなんともまぬけです。

以上のようなイベント処理を書いておけばまあなんとか用をなすでしょう。名前入力のウィンドウではシステムイベントを無視していますから，このウィンドウをオープンしている状態でウィンドウの全クローズ要求が来ても何も起きません（はっきりいって手抜きです）。

とにかく，このような方針で作成したプログラム（ほとんどTMSAMP.Sのマネという説もある）がリスト7です。リスト7のプログラムは_TmGetStrというサブルーチンになっていて，それをコールすると名前入力用のウィンドウがオープンし，文字入力が終わるとプログラム中の_TmGotTxtという領域に入力された文字列が，_TmGotLenという領域に入力された文字列のバイト数が格納されるようになっています。同時に_TmGotTxtのアドレスはレジスタa0に，文字列のバイト数はレジスタd0にも格納されるようになっています。このプログラムも別のプログラムに流用できるようにSXLIFEのソースからincludeするようになっています。

フォントパターンの画面への配置

サブルーチン_TmGetStrで得た文字列の各文字は適当にIOCSコールを使用すればROMのフォントのビットパターンに変換することができます。これはウィンドウを用いたプログラミングと直接は関係しないので詳細は省きますが，実際のプログラムはリスト8のようになっています。

リスト8ではメニュー項目の5番目(「文字列を入力」)が選択されてから，_TmGetStrで得た文字列（8文字のみ有効）のビットパターンをウィンドウ上に配置するまで

の全処理を示してあります。ウィンドウ上に表示する場合の文字の大きさのバランスを考えて，入力された半角文字はテーブルを引くことですべて対応する全角文字に変換するようにしてあるので注意してください。

＊

100%のSXLIFEではウィンドウ内の座標空間を4倍程度に広げるつもりだったのですが，元のプログラムをかなり最適化してあるため，あまりにも修正箇所が多くなって断念してしまいました。参考文献の「ライフゲイムの宇宙」内で紹介されている例題をすべて実行するためには空間の拡張がぜひとも必要だったのですが残念です。この点で完成したのは99%のSXLIFEといえるかもしれませんね。

ところで，旧版のSXLIFEのソースプログラム（謹賀新年PRO-68Kに付属）があるのをいいことに，前回，今回とプログラムの差分だけをリストとして掲載してきましたが，この差分リストから最新版のSXLIFEのソースプログラムを復元するのは少し困難かもしれませんね（ごめんなさい）。SXLIFEのソースを改造してみたけど思うように動かないという人はあと2カ月待ってみてください。5月号のおまけディスクに最新のSXLIFEが入る予定です。

《参考文献》

ウイリアム・パウンドストーン，「ライフゲイムの宇宙」，日本評論社，1990年．

リスト1　グローボックスの表示

```
 1:          .even
 2: drawGrowBox:
 3:          move.w  #%0011,-(sp)     ;; アクセスビットを設定し直す
 4:          .dc.w   __GMAPage
 5:          move.l  wPointer(a5),-(sp)
 6:          .dc.w   __WMDrawGBox     ;; グローボックスの描き直し
 7:          move.w  #%1111,-(sp)     ;; アクセスビットを元に戻す
 8:          .dc.w   __GMAPage
 9:          addq.l  #8,sp            ;; まとめてスタックを補正
10:          rts
```

リスト2　全クローズ処理

```
 1: * イベント構造（18バイト）
 2: what     equ     0                ;; イベントの種類
 3: with     equ     2                ;; イベントに関連した引数
 4: when     equ     6                ;; イベントの発生時（システム内部カウント）
 5: where    equ     10               ;; マウスの座標（グローバル座標系）
 6: addition equ     14               ;; 特殊キーの状態
 7: * タスクマンイベント一覧
 8: ENDTSK          equ     1        ;; タスクの終了
 9: CLOSEALL        equ     2        ;; 全ウィンドウのクローズ
10: WINDOWSELECT    equ     32       ;; ウィンドウのセレクト
11: ******************************************
12: * タスクマネージャイベント1、2の処理
13: ******************************************
14:          .even
15: EV_SYSTEM1:
16: EV_SYSTEM2:
17:          move.w  evntRec+addition(a5),d0
18:          cmp.w   #CLOSEALL,d0
19:          beq     SXfinish                ;; 処理を終了する
20:          cmp.w   #ENDTSK,d0
21:          beq     SXfinish                * 処理を終了する
22:          cmp.w   #WINDOWSELECT,d0
23:          bne     systemRet
24:          move.l  wPointer(a5),-(sp)
25:          .dc.w   __WMSelect
26:          addq.l  #4,sp
27: systemRet:
28:          rts
29:
30: SXfinish:
31:          lea.l   _SXterm(pc),a0
32:          move.l  a0,(sp)
33:          rts
```

リスト3　説明用ダイアログ

```
 1: * ファイル名:sxabout.s
 2: *
 3:          .even
 4: _AbTitle:
 5:          .dc.b   'このプログラムは・・・・',0
 6:
 7:          .even
 8: _About:
 9:          link    a6,#-4
10:          move.l  #-1,-(sp)
11:          clr.l   -(sp)
12:          move.w  #$f000,-(sp)         ;; 'DLOG' #$F000 （使用禁止のはず）
13:          .dc.w   __DMRefer            ;; ダイアログを出す
14:          lea     10(sp),sp
15:          move.l  a0,-4(a6)
16: *
17:          move.l  a0,-(sp)             ;; ダイアログの中を描く
18:          .dc.w   __GMSetGraph
19:          addq.l  #4,sp
20:          .dc.w   __GMInitPen
21:          move.w  #%0011,-(sp)
22:          .dc.w   __GMAPage
23:          addq.l  #2,sp
24: *
25:          move.l  #$0040_0004,-(sp)   ;; ウインドウのタイトル
26:          pea     _AbTitle(pc)
27:          .dc.w   __GMShadowStrZ
28:          addq.l  #8,sp
29: *
30:          move.l  #$0004_0020,-(sp)   ;; 作者
31:          pea     abMess1(pc)
32:          .dc.w   __GMShadowStrZ
33:          addq.l  #8,sp
34:          move.l  #$0004_0038,-(sp)   ;; コメント
35:          pea     abMess2(pc)
36:          .dc.w   __GMShadowStrZ
37:          addq.l  #8,sp
38:          move.l  #$0040_0050,-(sp)   ;; バージョン
39:          pea     abMess3(pc)
40:          .dc.w   __GMShadowStrZ
41:          addq.l  #8,sp
42: *
43:          clr.l   -(sp)
44:          .dc.w   __DMControl                 ;; ボタンが押されるのを待つ
45:          addq.l  #4,sp
46: *
47:          move.l  -4(a6),-(sp)
48:          .dc.w   __DMDispose                 ;; ダイアログを消す
49:          addq.l  #4,sp
50: *
51:          unlk    a6
52:          rts
53:
```

リスト4　デモプログラム

```
 1: * ファイル名:sxdemo.s
 2: *
 3: * d0.w にデモ番号を入れてコール
 4: *
 5:          .even
 6: demoLife:
 7:          movem.l d3-d7/a3,-(sp)
 8: *
 9:          moveq.l #0,d3
10:          moveq.l #0,d6
11:          moveq.l #0,d7
12:          lea     _field(a5),a3
13:          move.l  a3,d4
14:          ext.l   d0
15:          add.w   d0,d0
16:          lea     demoOrg(pc),a3
17:          move.w  (a3,d0.w),d0      ;; オフセット
18:          lea     (a3,d0.w),a3
19:          move.b  (a3)+,d6          ;; 垂直ベース
20:          move.b  (a3)+,d7          ;; 水平ベース
21: demoLoop:
22:          move.b  (a3,d3.l),d0
23:          ext.w   d0
24:          move.w  d0,d1
25:          ext.l   d1
26:          move.b 1(a3,d3.l),d0
27:          ext.w   d0
28:          move.w  d0,a1
29:          addq.l  #2,d3
30:          cmp.w   #0,a1
31:          blt     demoExit
32:          add.l   d6,d1
33:          adda.l  d7,a1
34: *
35:          move.l  d1,d0              ;; メモリにドット設定
36:          asl.l   #6,d0
37:          move.l  d4,a0
38:          add.l   d0,a0
39:          move.b  #1,(a1,a0.l)
40: *
41:          move.w  a1,-(sp)           ;; 画面にドット設定
42:          move.w  d1,-(sp)
43:          bsr     pset
44:          addq.l  #4,sp
45: *
46:          bra demoLoop
47: demoExit:
48:          movem.l (sp)+,d3-d7/a3
49:          rts
50:
```

```
 51:          .even
 52: demoOrg:
 53:          .dc.w    _demo1-demoOrg
 54:          .dc.w    _demo2-demoOrg
 55:          .dc.w    _demo3-demoOrg
 56:          .dc.w    _demo4-demoOrg
 57:          .dc.w    _demo5-demoOrg
 58:          .dc.w    _demo6-demoOrg
 59:          .dc.w    _demo7-demoOrg
 60:          .dc.w    _demo8-demoOrg
 61:          .dc.w    _demo9-demoOrg
 62:          .dc.w    _demo10-demoOrg
 63:          .dc.w    _demo11-demoOrg
 64:          .dc.w    _demo12-demoOrg
 65:          .dc.w    _demo13-demoOrg
 66:          .dc.w    _demo14-demoOrg
 67:          .dc.w    _demo15-demoOrg
 68:          .dc.w    _demo16-demoOrg
 69:
 70: *
 71: * ライフゲームのデモパターン（縦：40，横：64）
 72: *
 73: * データ構造    .dc.b   V,H                 （原点）
 74: *                .dc.b   V0,H0,････         （存在点：相対座標）
 75: *                .dc.b   $ff,$ff            （終了コード）
 76:          .even
 77: *
 78: * ブリンカー
 79: *
 80: _demo1:
 81:          .dc.b    5,5
 82:          .dc.b    0,0,0,1,0,2
 83:          .dc.w    -1
 84:
 85:          .even
 86: *
 87: * Tテトロミノ
 88: *
 89: _demo2:
 90:          .dc.b    5,5
 91:          .dc.b    0,0,0,1,0,2,1,1
 92:          .dc.w    -1
 93:
 94:          .even
 95: *
 96: * グライダー
 97: *
 98: _demo3:
 99:          .dc.b    5,5
100:          .dc.b    2,0,2,1,2,2,1,2,0,1
101:          .dc.w    -1
102:
103:          .even
104: *
105: * Rペントミノ
106: *
107: _demo4:
108:          .dc.b    20,30
109:          .dc.b    1,0,1,1,0,1,2,1,0,2
110:          .dc.w    -1
111:
112:          .even
113: *
114: * πペントミノ
115: *
116: _demo5:
117:          .dc.b    20,20
118:          .dc.b    0,0,0,1,0,2,1,0,1,2,2,0,2,2
119:          .dc.w    -1
120:
121:          .even
122: *
123: * 世紀
124: *
125: _demo6:
126:          .dc.b    20,20
127:          .dc.b    0,2,1,1,1,2,2,0,2,1,3,1
128:          .dc.w    -1
129:
130:          .even
131: *
132: * パルサー
133: *
134: _demo7:
135:          .dc.b    5,5
136:          .dc.b    0,2,0,3,0,4,2,0,3,0,4,0
137:          .dc.b    2,5,3,5,4,5,5,2,5,3,5,4
138:          .dc.b    0,8,0,9,0,10,2,7,3,7,4,7
139:          .dc.b    2,12,3,12,4,12,5,8,5,9,5,10
140:          .dc.b    7,2,7,3,7,4,8,0,9,0,10,0
141:          .dc.b    8,5,9,5,10,5,12,2,12,3,12,4
142:          .dc.b    7,8,7,9,7,10,8,7,9,7,10,7
143:          .dc.b    8,12,9,12,10,12,12,8,12,9,12,10
144:          .dc.w    -1
145:
146:          .even
147: *
148: * 8の字
149: *
150: _demo8:
151:          .dc.b    5,5
152:          .dc.b    0,0,0,1,0,2,1,0,1,1,1,2,2,0,2,1,2,2
153:          .dc.b    3,3,3,4,3,5,4,3,4,4,4,5,5,3,5,4,5,5
154:          .dc.w    -1
155:
156:          .even
157: *
158: * 理髪店の看板
159: *
160: _demo9:
161:          .dc.b    5,5
162:          .dc.b    0,9,0,10,1,10,2,7,2,9,4,5,4,7,6,3,6,5
163:          .dc.b    8,1,8,3,9,0,10,0,10,1
164:          .dc.w    -1
165:
166:          .even
167: *
168: * フリップフロップ
169: *
170: _demo10:
171:          .dc.b    5,5
172:          .dc.b    0,3,1,3,1,5,2,1,3,6,3,7,4,0,4,1,5,6
173:          .dc.b    6,2,6,4,7,4
174:          .dc.w    -1
175:
176:          .even
177: *
178: * 銀河
179: *
180: _demo11:
181:          .dc.b    5,5
182:          .dc.b    0,0,0,1,0,2,0,3,0,4,0,5
183:          .dc.b    1,0,1,1,1,2,1,3,1,4,1,5
184:          .dc.b    3,0,3,1,4,0,4,1,5,0,5,1
185:          .dc.b    6,0,6,1,7,0,7,1,8,0,8,1
186:          .dc.b    0,7,0,8,1,7,1,8,2,7,2,8
187:          .dc.b    3,7,3,8,4,7,4,8,5,7,5,8
188:          .dc.b    7,3,7,4,7,5,7,6,7,7,7,8
189:          .dc.b    8,3,8,4,8,5,8,6,8,7,8,8
190:          .dc.w    -1
191:
192:          .even
193: *
194: * タンブラー
195: *
196: _demo12:
197:          .dc.b    5,5
198:          .dc.b    0,1,0,2,0,4,0,5,1,1,1,2,1,4,1,5
199:          .dc.b    2,2,2,4,3,2,3,4,3,0,3,6,4,0,4,2
200:          .dc.b    4,4,4,6,5,0,5,1,5,5,5,6
201:          .dc.w    -1
202:
203:          .even
204: *
205: * 時計 II（昔のパターンを流用しているので見にくい）
206: *
207: _demo13:
208:          .dc.b    5,5
209:          .dc.b    19-13,13-13,20-13,13-13
210:          .dc.b    19-13,14-13,20-13,14-13
211:          .dc.b    17-13,16-13,18-13,16-13
212:          .dc.b    19-13,16-13,20-13,16-13
213:          .dc.b    13-13,17-13,14-13,17-13
214:          .dc.b    16-13,17-13,21-13,17-13
215:          .dc.b    13-13,18-13,14-13,18-13
216:          .dc.b    16-13,18-13,17-13,18-13
217:          .dc.b    21-13,18-13,16-13,19-13
218:          .dc.b    19-13,19-13,21-13,19-13
219:          .dc.b    23-13,19-13,24-13,19-13
220:          .dc.b    16-13,20-13.18-13,20-13
221:          .dc.b    21-13,20-13,23-13,20-13
222:          .dc.b    24-13,20-13,17-13,21-13
223:          .dc.b    18-13,21-13,19-13,21-13
224:          .dc.b    20-13,21-13,17-13,23-13
225:          .dc.b    18-13,23-13,17-13,24-13
226:          .dc.b    18-13,24-13
227:          .dc.w    -1
228:
229:          .even
230: *
231: * ペンタデカソロン+グライダー
232: *
233: _demo14:
234:          .dc.b    5,5
235:          .dc.b    0,27,0,30,0,35,0,38,1,25,1,26,1,27,1,30,1,31,1,32
236:          .dc.b    1,33,1,34,1,35,1,38,1,39,1,40,2,27,2,30,2,35,2,38
237:          .dc.b    4,2,4,5,4,10,4,13,5,0,5,1,5,2,5,5,5,6,5,7,5,8,5,9
238:          .dc.b    5,10,5,13,5,14,5,15,5,18,6,2,6,5,6,10,6,13,6,17
239:          .dc.b    7,17,7,18,7,19
240:          .dc.w    -1
241:
242:          .even
243: *
244: * エデンの園
245: *
246: _demo15:
247:          .dc.b    3,30
248:          .dc.b    0,0,0,1,0,3,0,5,0,6,0,7,0,8
249:          .dc.b    1,0,1,2,1,3,1,4,1,6,1,8
250:          .dc.b    2,0,2,1,2,3,2,5,2,6,2,7,2,8
251:          .dc.b    3,0,3,2,3,3,3,4,3,6,3,8
252:          .dc.b    4,0,4,1,4,3,4,5,4,6,4,7,4,8
253:          .dc.b    5,0,5,2,5,3,5,4,5,6,5,8
254:          .dc.b    6,0,6,1,6,3,6,5,6,6,6,7,6,8
255:          .dc.b    7,0,7,2,7,3,7,4,7,6,7,8
256:          .dc.b    8,0,8,1,8,3,8,5,8,6,8,7,8,8
257:          .dc.b    9,0,9,2,9,3,9,4,9,6,9,8
258:          .dc.b    10,0,10,1,10,3,10,5,10,6,10,7,10,8
259:          .dc.b    11,0,11,2,11,3,11,4,11,6,11,8
260:          .dc.b    12,0,12,1,12,3,12,5,12,6,12,7,12,8
261:          .dc.b    13,0,13,2,13,3,13,4,13,5,13,7,13,8
262:          .dc.b    14,0,14,1,14,2,14,4,14,6,14,7
263:          .dc.b    15,0,15,2,15,3,15,4,15,5,15,7,15,8
264:          .dc.b    16,0,16,1,16,3,16,5,16,6,16,8
265:          .dc.b    17,0,17,2,17,3,17,4,17,5,17,7,17,8
266:          .dc.b    18,0,18,1,18,2,18,3,18,5,18,6,18,7,18,8
267:          .dc.b    19,0,19,1,19,2,19,4,19,5,19,6,19,8
268:          .dc.b    20,0,20,2,20,3,20,4,20,6,20,7,20,8
269:          .dc.b    21,0,21,1,21,3,21,4,21,5,21,7,21,8
270:          .dc.b    22,0,22,1,22,2,22,3,22,5,22,6,22,7,22,8
271:          .dc.b    23,0,23,1,23,2,23,4,23,5,23,6,23,8
272:          .dc.b    24,0,24,2,24,3,24,4,24,6,24,7,24,8
273:          .dc.b    25,0,25,1,25,3,25,4,25,5,25,7,25,8
274:          .dc.b    26,0,26,1,26,2,26,3,26,5,26,6,26,7,26,8
275:          .dc.b    27,0,27,1,27,2,27,4,27,5,27,6,27,8
276:          .dc.b    28,0,28,2,28,3,28,4,28,6,28,7,28,8
277:          .dc.b    29,0,29,1,29,3,29,5,29,7,29,8
278:          .dc.b    30,0,30,2,30,3,30,4,30,5,30,6,30,8
279:          .dc.b    31,0,31,1,31,3,31,5,31,6,31,7,31,8
280:          .dc.b    32,0,32,1,32,2,32,3,32,4,32,5,32,7,32,8
```

▶学費値上げ反対のストで試験が4月になってしまいました。気が抜けてしまって毎日ぼーっとしています。
岸田　英之(20)東京都

```
281:          .dc.w    -1
282:
283:          .even
284: *
285: * 宇宙の熊手
286: *
287: _demo16:
288:          .dc.b    8,1
289:          .dc.b    0,0,0,1,0,2,1,2,2,1,5,5,5,6,5,7,6,7,7,6
290:          .dc.b    10,10,10,11,10,12,11,12,12,11
291:          .dc.b    15,15,15,16,15,17,16,17,17,16
292:          .dc.b    18,21,19,21,18,23,19,24,21,22,21,23,21,24
293:          .dc.b    23,25,23,28,25,29,26,26,26,27,26,28,26,29
294:          .dc.b    8,36,8,37,9,34,9,35,9,37,9,38
295:          .dc.b    10,34,10,35,10,36,10,37,11,35,11,36
296:          .dc.b    8,43,8,44,8,45,8,46,9,42,9,46,10,46,11,42,11,45
297:          .dc.b    13,41,14,36,14,40,14,43,15,35,15,36,15,44,16,36
298:          .dc.b    16,37,16,44,17,37,17,38,17,43,17,44,18,42
299:          .dc.b    19,39,19,40,22,43,22,44,22,45,22,46,23,42,23,46
300:          .dc.b    24,46,25,42,25,45
301:          .dc.w    -1
```

リスト5 ライフゲームの初期化

```
 1:          .even
 2: initLife:
 3:          movem.l  d1/a1,-(sp)
 4:          move.w   #1023,d1          ;; ループ回数
 5:          lea      _field(a5),a1
 6: LL9:
 7:          clr.l    (a1)+             ;; メモリを初期化
 8:          dbra     d1,LL9
 9: *
10:          bsr      initMap           ;; 画面をクリア
11: *
12:          moveq.l  #$56,d0           ;; 現在の時刻を得て
13:          trap     #15
14:          move.l   d0,d1
15:          moveq.l  #$57,d0           ;; それを2進データに変換し
16:          trap     #15
17:          and.l    #$f,d0            ;; デモの番号とする
18: *
19:          bsr      demoLife          ;; 適当なデモを実行
20: *
21:          movem.l  (sp)+,d1/a1
22:          rts
```

リスト6 ドット設定ボタンの変更

```
  1: EV_MSLDOWN:
  2:          movem.l d1-d7/a1-a5,-(sp)     ;; レジスタを保存
  3: *
  4: * 自分のウインドウかを調べる
  5: *
  6:          move.l   evntRec+with(a5),d0
  7:          beq      retMSLDOWN           ;; どのウインドウにも該当しない
  8:          cmp.l    wPointer(a5),d0
  9:          bne      retMSLDOWN           ;; 他のウインドウだった
 10: *
 11: * カレントポートにセット
 12: *
 13:          move.l   d0,-(sp)
 14:          .dc.w    __GMSetGraph
 15:          addq.l   #4,sp
 16:          move.l   a0,a2
 17:          tst.w    active(a5)
 18:          bne      procMSLDOWN0         ;; 前からアクティブだった
 19:          move.w   #1,active(a5)
 20:          move.w   #$ffff,evntMsk(a5)   ;; イベントマスクをセット
 21: *
 22: * ウインドウを切り替える
 23: *
 24:          pea      (a2)
 25:          .dc.w    __WMSelect           ;; 自分をセレクト
 26:          addq.l   #4,sp
 27:          .dc.w    __EMLStill           ;; 左ボタンが押されたままか
 28:          tst.l    d0
 29:          beq      noStillMSLDOWN
 30:          move.l   evntRec+where(a5),-(sp)
 31:          .dc.w    __WMFind             ;; 座標がウインドウのどこにあるか
 32:          addq.l   #4,sp
 33:          cmp.w    #inDrag,d0
 34:          beq      procMSLDOWN
 35:          cmp.w    #inGrow,d0
 36:          beq      procMSLDOWN
 37:          cmp.w    #inContent,d0
 38:          bne      noStillMSLDOWN       ;; コンテントリージョン以外
 39: *
 40: * ここではコンテントリージョン(除くグローボックス)
 41: *
 42: procMSLDOWN0:
 43: .ifndef _RightSet_
 44:          tst.b    userWRK(a5)          ;; 設定モードか
 45:          bne      dotSetReset
 46: .endif
 47: *
 48: * いよいよウインドウマネージャーを呼ぶ
 49: *
 50: procMSLDOWN:
 51:          pea      evntRec(a5)
 52:          pea      (a2)                 ;; ウインドーレコードへのポインタ
 53:          .dc.w    __SXCallWindM        ;; 移動等の処理を行う
 54:          addq.l   #8,sp
 55:          tst.l    d0
 56:          bmi      errMSLDOWN           ;; エラー
 57: * サイズ固定
 58:          tst.w    d0
 59:          beq      noStillMSLDOWN
 60:          cmp.w    #inZoomIn,d0
 61:          beq      procClip
 62:          cmp.w    #inZoomOut,d0
 63:          beq      procClip
 64:          cmp.w    #inGrow,d0
 65:          beq      procClip
 66:          cmp.w    #inGoAway,d0
 67:          beq      _SXgoAway
 68: *
 69: * イベントレコードを除く
 70: *
 71: noStillMSLDOWN:
 72:          pea      evntRec(a5)          ;; イベントレコード
 73:          move.w   evntMsk(a5),-(sp)    ;; イベントマスク
 74:          .dc.w    __TSGetEvent
 75:          addq.l   #6,sp
 76: *
 77: * 今回の状態をセーブ
 78: retMSLDOWN:
 79: .ifdef _RightSet_
 80:          clr.b    userWRK(a5)          ;; 描画のロックを解除
 81: .endif
 82:          movem.l  (sp)+,d1-d7/a1-a5
 83:          rts
 84: ****************************************************************
 85: procMSRDOWN:
 86: .ifdef _RightSet_
 87:          tst.b    userWRK(a5)          ;; 設定モードか
 88:          bne      dotSetReset
 89: .endif
 90: **
 91:          move.l   evntRec+where(a5),-(sp)
 92:          move.l   _menuHdl(a5),-(sp)
 93:          .dc.w    __MNSelect
 94:          addq.l   #8,sp
 95:          tst.b    d0
 96:          beq      doMenu0
 97:          cmpi.b   #3,d0
 98:          blt      doMenu1
 99:          beq      doMenu3
100:          cmpi.b   #5,d0
101:          blt      doMenu4
102:          beq      doMenu5
103:          bgt      doDemos
104:          bra      noAction
105:
106: ***************************************************************
107: doMenu4:
108: .ifdef _RightSet_
109:          move.b   #1,userWRK(a5)       ;; 描画をロックする
110:          bra      noAction             ;; ロックの解除は左ボタン
111: .else
112:          movea.l  _menuHdl(a5),a2
113:          movea.l  (a2),a2
114:          lea      (togglm-mProt+15)(a2),a2   ;; 変更項目の先頭
115:          tst.b    userWRK(a5)
116:          beq      startDSet
117: endDSet:
118:          lea      mm0(pc),a0
119:          clr.b    userWRK(a5)
120:          bra      doM4CMN
121: startDSet:
122:          lea      mm1(pc),a0
123:          move.b   #1,userWRK(A5)
124: doM4CMN:
125:          move.b   (a0)+,(a2)+
126:          move.b   (a0)+,(a2)+
127:          move.b   (a0)+,(a2)+
128:          move.b   (a0)+,(a2)+
129:          bra      noAction
130: mm0:
131:          .dc.b    '開始'
132: mm1:
133:          .dc.b    '終了'
134: .endif
```

リスト7 名前の入力

```
 1: * ファイル名:getstr.s
 2: *
 3: **************************************************
 4: *
 5: * これは汎用の1行入力ウインドウを実現する
 6: *
 7: **************************************************
 8: _TmWinPtr       equ      -4
 9: _TmTEHdl        equ      -8
10: _TmActive       equ      -10
11: _TmBound        equ      -18
12: _TmEvMsk        equ      -20
13: _TmEvRec        equ      -38
14: _TmParam        equ      -40
15:                 .text
16:
17: .offset 0
18: teDestRect:     ds.w     4        ;; 内部で使用
19: teViewRect:     ds.w     4        ;; ビューレクタングル
20: teDandVRect:    ds.w     4        ;; 内部で使用
```

▶亀田さん，私の誕生日も同じです。おかげでもらえる年齢層の高いこと高いこと。ムナシイ。周りには同じ誕生日の人が3人います。しかし，女……。　清水　雅士(17)山口県

```
 21: teHText:        ds.l    1          ;; 編集テキストへのハンドル
 22: teLenMax:       ds.l    1          ;; 編集テキストの入力最大数
 23: teLength:       ds.l    1          ;; 既に入力されている編集テキストの長さ
 24: teRsv0:         ds.l    1          ;; 内部で使用
 25: teSelStart:     ds.l    1          ;; セレクト範囲の開始位置
 26: teSelEnd:       ds.l    1          ;; セレクト範囲の終了位置
 27: teSelLine:      ds.l    1          ;; 現在のセレクト行位置
 28: teSeloffSet:    ds.l    1          ;; 現在のバッファのセレクト位置
 29: teRsv1:         ds.l    1          ;; 内部で使用
 30: teLineHeight:   ds.w    1          ;; 改行幅
 31: teTabSize:      ds.w    1          ;; ＴＡＢサイズ（ドット単位）
 32: teJust:         ds.w    1          ;; 行揃え（0:左寄せ 1:中央寄せ -1:右寄せ）
 33: teRsv2:         ds.w    5          ;; 内部で使用
 34: teInPort:       ds.l    1          ;; <graph> へのポインタ
 35: teCaretTime:    ds.l    1          ;; 内部で使用
 36: teCaretState:   ds.w    1          ;; 内部で使用
 37: teRsv3:         ds.l    1          ;; 内部で使用
 38: teRsv4:         ds.l    1          ;; 内部で使用
 39: teRsv5:         ds.l    1          ;; 内部で使用
 40: teRsv6:         ds.l    1          ;; 内部で使用
 41: teRsv7:         ds.l    1          ;; 内部で使用
 42: teNLines:       ds.l    1          ;; テキストの行数
 43: teLineStarts:   ds.l    1          ;; 内部で使用
 44: tEdit
 45:         .text
 46:
 47:         .even
 48: _TmWinTitle:
 49:         .dc.b   14,'なにか入力して'
 50:
 51:         .even
 52: _TmdRect:
 53:         .dc.w   6*2,4,6*256,20
 54: _TmvRect:
 55:         .dc.w   6*2,4,300-6*2,20
 56:
 57: _TmGotTxt:
 58:         .ds.b   512
 59: _TmgotLen:
 60:         .dc.l   0
 61:
 62:         .even
 63: _TmGetStr:
 64:         link    a6,#_TmParam
 65:         movem.l d1-d7/a1-a2,-(sp)    ;; レジスタを保存
 66: *
 67:         clr.l   _TmWinPtr(a6)        ;; ウィンドウレコードへのポインタ
 68:         clr.l   _TmTEHdl(a6)         ;; <tEdit> のハンドルをクリアする
 69:         clr.w   _TmActive(a6)        ;; アクテイブの時にオン
 70: *
 71: * ウィンドウの位置と大きさを計算する
 72: *
 73:         .dc.w   __TSGetWindowPos     ;; ウインドウ開始位置
 74:         move.l  d0,_TmBound(a6)
 75:         add.l   #$012c0014,d0        ;; ウインドウの大きさ
 76:         move.l  d0,_TmBound+4(a6)
 77: *
 78: * 自分のタスクＩＤをウィンドウに登録する
 79: *
 80:         .dc.w   __TSGetID            ;; 自分のタスクＩＤを得る
 81: *
 82: * ウィンドウをオープンする
 83: *
 84:         move.l  d0,-(sp)             ;; タスクＩＤ
 85:         move.w  #$ffff,-(sp)         ;; クローズボックスあり
 86:         move.l  #$ffffffff,-(sp)     ;; 一番手前
 87:         move.w  #$200,-(sp)          ;; 定義関数のＩＤ
 88:         move.w  #$ffff,-(sp)         ;; ウィンドウを表示する
 89:         pea     _TmWinTitle(pc)      ;; タイトル
 90:         pea     _TmBound(a6)         ;; ウィンドウサイズ
 91:         clr.l   -(sp)                ;; メモリを確保する
 92:         .dc.w   __WMOpen
 93:         lea     26(sp),sp
 94:         tst.l   d0
 95:         bmi     err_Open             ;; オープンできなかった
 96:         move.l  a0,_TmWinPtr(a6)     ;; ウィンドウレコードへのポインタをセーブ
 97:         pea     (a0)
 98:         .dc.w   __GMSetGraph         * <current graph> にセット
 99:         addq.l  #4,sp
100: *
101: * アクセスビットをセット
102: *
103:         move.w  #%0011,-(sp)
104:         .dc.w   __GMAPage
105:         addq.l  #2,sp
106: *
107: * フォアグランドカラーをセット
108: *
109:         move.w  #%1011,-(sp)
110:         .dc.w   __GMForeColor
111:         addq.l  #2,sp
112: *
113: * バックグランドカラーをセット
114: *
115:         move.w  #%1001,-(sp)
116:         .dc.w   __GMBackColor
117:         addq.l  #2,sp
118: *
119: * 文字の属性をセット
120: *
121:         move.w  #0,-(sp)                 ;; 6
122:         .dc.w   __GMFontKind
123:         addq.l  #2,sp
124: *
125: * テキストエディトのレコードを作成する
126: *
127:         move.l  _TmWinPtr(a6),-(sp)
128:         pea     _TmvRect(pc)             ;; view
129:         pea     _TmdRect(pc)             ;; dest
130:         .dc.w   __TMNew2
131:         lea     12(sp),sp
132:         bmi     err__TmTEHdl             ;; エラー
133:         move.l  a0,_TmTEHdl(a6)          ;; ハンドルをセーブする
134:         move.l  (a0),a0
135:         move.l  #256+88,teLenMax(a0)     * 最大文字数
136:         move.w  #12,teLineHeight(a0)     * 改行幅
137: *
138: * 該当するイベントのタイプを指定する
139: *
140:         move.w  #%0011_0010_1000_0010,_TmEvMsk(a6)
141: *
142: * イベント処理ループ
143: *
144: _TmEvLoop:
145:         pea     _TmEvRec(a6)             ;; イベントレコード
146:         move.w  _TmEvMsk(a6),-(sp)       ;; イベントマスク
147:         .dc.w   __TSEventAvail           ;; タスクを切り替える
148:         addq.l  #6,sp
149: *
150: * イベント発生時の処理を呼び出す
151: *
152:         move.w  _TmEvRec+0(a6),d0        ;; 発生したイベント
153:         and.w   #$000f,d0
154:         beq     _TmEvNull
155:         cmpi.b  #1,d0
156:         beq     _TmEvMSLDWN
157:         cmpi.b  #3,d0
158:         beq     _TmEvMSRDWN
159:         cmpi.b  #5,d0
160:         beq     _TmEvKey
161:         cmpi.b  #7,d0
162:         beq     _TmEvUpdate
163:         cmpi.b  #9,d0
164:         beq     _TmEv_TmActive
165:         and.b   #$fe,d0
166:         cmpi.b  #12,d0
167:         beq     _TmEvSys12
168:         bra     _TmEvLoop
169: *********************************************
170: *
171: * Null Event Processing
172: *
173: _TmEvNull:
174:         move.l  _TmWinPtr(a6),-(sp)
175:         .dc.w   __GMSetGraph
176:         addq.l  #4,sp
177: *
178: * テキストマンを呼び出す
179: *
180:         pea     _TmEvRec(a6)
181:         move.l  _TmTEHdl(a6),-(sp)
182:         .dc.w   __TMEvent
183:         addq.l  #8,sp
184:         bra     _TmEvLoop
185: *******************************************
186: *
187: * Mouse Left Button Down Processing
188: *
189: _TmEvMSLDWN:
190: *
191: * 自分のウィンドウかを調べる
192: *
193:         move.l  _TmWinPtr(a6),a2
194:         cmp.l   _TmEvRec+2(a6),a2
195:         bne     _TmEvLoop                ;; 他のウィンドウだった
196:         tst.w   _TmActive(a6)
197:         bne     _TmMslCallWM             ;; 前からアクティブだった
198: *
199: * ウィンドウを切り替える
200: *
201:         pea     (a2)
202:         .dc.w   __WMSelect               ;; ウィンドウを切り替える
203:         addq.l  #4,sp
204:         move.l  _TmEvRec+10(a6),-(sp)
205:         .dc.w   __WMFind                 ;; マウスの位置
206:         addq.l  #4,sp
207:         cmp.w   #inContent,d0
208:         beq     _TmMslCallTEM            ;; コンテントリージョンの中
209:         cmp.w   #inDrag,d0
210:         bne     _TmMslNextEv             ;; タイトルバーの上ではない
211:         .dc.w   __EMLStill
212:         beq     _TmMslNextEv             ;; マウスがはなされた
213: *
214: * ウィンドウマンを呼び出す
215: *
216: _TmMslCallWM:
217:         pea     _TmEvRec(a6)
218:         pea     (a2)                     ;; ウィンドウレコードへのポインタ
219:         .dc.w   __SXCallWindM
220:         addq.l  #8,sp
221:         cmp.w   #inClose,d0              ;; クローズボックスが押されたら
222:         beq     _TmRetKey                ;; 終了する
223: *
224: * テキストマンを呼び出す
225: *
226: _TmMslCallTEM:
227:         move.l  _TmWinPtr(a6),-(sp)
228:         .dc.w   __GMSetGraph             * <current graph> にセット
229:         addq.l  #4,sp
230: *
231: * <tEdit> レコードのビューレクタングルに入っているかを調べる
232: *
233:         .dc.w   __EMMSLoc                ;; マウスのローカル座標を得る
234:         move.l  d0,-(sp)
235:         move.l  _TmTEHdl(a6),a0
236:         move.l  (a0),a0
237:         pea     teViewRect(a0)           ;; ビューレクタングル
238:         .dc.w   __GMPtInRect
239:         addq.l  #8,sp
240:         beq     _TmMslNextEv
241:         pea     _TmEvRec(a6)
242:         move.l  _TmTEHdl(a6),-(sp)
243:         .dc.w   __TMEvent
244:         addq.l  #8,sp
```


▶「歩く城」とでもいいましょうか。なにか哀愁を感じさせる２月号の表紙。私は初めて見たとき，「顔」のように見えてしまった。　矢野　慎一(19)埼玉県

```
245: *
246: * イベントレコードを取りのぞく
247: *
248: _TmMslNextEv:
249:         pea     _TmEvRec(a6)            ;; イベントレコード
250:         move.w  _TmEvMsk(a6),-(sp)      ;; イベントマスク
251:         .dc.w   __TSGetEvent            ;; イベントを除く
252:         addq.l  #6,sp
253:         bra     _TmEvLoop
254: ******************************************
255: *
256: * Mouse Right Button Down Processing
257: *
258: _TmEvMSRDWN:
259: *
260: * 自分のウィンドウかを調べる
261: *
262:         move.l  _TmEvRec+2(a6),d0
263:         cmp.l   _TmWinPtr(a6),d0
264:         bne     _TmEvLoop               ;; 他のウィンドウだった
265: *
266: * イベントを取り除く
267: *
268:         pea     _TmEvRec(a6)            ;; イベントレコード
269:         move.w  _TmEvMsk(a6),-(sp)      ;; イベントマスク
270:         .dc.w   __TSGetEvent
271:         addq.l  #6,sp
272: *
273: * <current graph> にセット
274: *
275:         move.l  _TmWinPtr(a6),-(sp)
276:         .dc.w   __GMSetGraph
277:         addq.l  #4,sp
278: *
279: * テキストマンを呼び出す
280: *
281:         pea     _TmEvRec(a6)
282:         move.l  _TmTEHdl(a6),-(sp)
283:         .dc.w   __TMEvent
284:         addq.l  #8,sp
285:         bra     _TmEvLoop
286: ******************************************
287: *
288: * Key Down Processing
289: *
290: _TmEvKey:
291:         tst.w   _TmActive(a6)
292:         beq     _TmEvLoop               ;; アクティブではない
293: *
294: * イベントを取り除く
295: *
296:         pea     _TmEvRec(a6)            ;; イベントレコード
297:         move.w  _TmEvMsk(a6),-(sp)      ;; イベントマスク
298:         .dc.w   __TSGetEvent
299:         addq.l  #6,sp
300:         cmp.w   #$0d,_TmEvRec+4(a6)
301:         bne     keyProc
302:
303: ******************************************
304: * リターンキーの処理
305: ******************************************
306: *
307: * テキストを得る
308: *
309: _TmRetKey:
310:         move.l  #512,-(sp)
311:         pea     _TmGotTxt(pc)
312:         move.l  _TmTEHdl(a6),-(sp)
313:         .dc.w   __TMGetText
314:         lea     12(sp),sp
315:         lea     _TmgotLen(pc),a0
316:         move.l  d0,(a0)
317:         bra     _TmEndPr
318:
319: ******************************************
320: * キー入力処理
321: ******************************************
322: keyProc:
323: *
324: * <current graph> にセット
325: *
326:         move.l  _TmWinPtr(a6),-(sp)
327:         .dc.w   __GMSetGraph
328:         addq.l  #4,sp
329: *
330: * テキストマンを呼び出す
331: *
332:         pea     _TmEvRec(a6)
333:         move.l  _TmTEHdl(a6),-(sp)
334:         .dc.w   __TMEvent
335:         addq.l  #8,sp
336:         bra     _TmEvLoop
337: *********************************************
338: *
339: * Update Processing
340: *
341: _TmMasterUpd:
342:         bsr     EV_UPDATE
343:         bra     _TmEvLoop
344:
345: _TmEvUpdate:
346: *
347: * 自分のウィンドウかを調べる
348: *
349:         move.l  _TmEvRec+2(a6),d0
350:         cmp.l   wPointer(a5),d0
351:         beq     _TmMasterUpd
352:         cmp.l   _TmWinPtr(a6),d0        ;; ウィンドウレコードへのポインタ
353:         bne     _TmEvLoop               ;; 自分のウィンドウじゃない
354: *
355: * <current graph> にセット
356: *
357:         move.l  _TmWinPtr(a6),-(sp)
358:         .dc.w   __GMSetGraph
359:         addq.l  #4,sp
360: *
361: * カーソルを消す (WMUpdate() をコールする前に必ずカーソルを消す)
362: *
363:         clr.w   -(sp)
364:         move.l  _TmTEHdl(a6),-(sp)      ;; ハンドル
365:         .dc.w   __TMCaret
366:         addq.l  #6,sp
367: *
368: * クリップリージョンをセット
369: *
370:         move.l  _TmWinPtr(a6),-(sp)
371:         .dc.w   __WMUpdate
372: *
373: * 画面を再表示する
374: *
375: *
376: * <current graph> にセット
377: *
378:         move.l  _TmWinPtr(a6),-(sp)
379:         .dc.w   __GMSetGraph
380:         addq.l  #4,sp
381: *
382: * テキストを表示する
383: *
384:         pea     _TmvRect(pc)            ;; 囲む四角形
385:         move.l  _TmTEHdl(a6),-(sp)      ;; <tEdit> レコードへのハンドル
386:         .dc.w   __TMUpDate
387:         addq.l  #8,sp
388: *
389: * クリップリージョンを戻す
390: *
391:         .dc.w   __WMUpdtOver
392:         addq.l  #4,sp
393:         bra     _TmEvLoop
394: ************************************************
395: *
396: * Activate Processing
397: *
398: _TmEv_TmActive:
399: *
400: * 自分のウィンドウかを調べる
401: *
402:         move.l  _TmEvRec+2(a6),d0
403:         beq     _TmEvLoop               ;; どのウィンドウにも該当しない
404:         cmp.l   _TmWinPtr(a6),d0        ;; ウィンドウレコードへのポインタ
405:         bne     inact                   ;; 自分のウィンドウじゃない
406: *
407: * 自分のウィンドウがアクティブになった
408: *
409:         move.w  #1,_TmActive(a6)        ;; アクティブフラグをセット
410: *
411: * 該当するイベントのタイプを指定する
412: *
413:         move.w  #%0011_0010_1010_1011,_TmEvMsk(a6)
414:         bra     _TmEvLoop
415: *
416: * 他のウィンドウがアクティブになった
417: *
418: inact:
419:         clr.w   _TmActive(a6)
420: *
421: * 該当するイベントのタイプを指定する
422: *
423:         move.w  #%0011_0010_1000_0010,_TmEvMsk(a6)
424:         bra     _TmEvLoop
425: *********************************************
426: *
427: * System Event Processing
428: *
429: _TmEvSys12:
430:         bra     _TmEvLoop
431: *********************************************
432: err_Open:
433:         moveq.l #-1,d0
434:         bra     _TmRet2
435:
436: err__TmTEHdl:
437:         move.l  _TmWinPtr(a6),d0        ;; ウィンドウレコードへのポインタ
438:         beq     _TmRet                  ;; ウィンドウがない
439:         move.l  d0,-(sp)
440:         .dc.w   __WMDispose             ;; ウィンドウを廃棄
441:         addq.l  #4,sp
442:         bra     _TmRet2
443:
444: _TmEndPr:
445:         move.l  _TmWinPtr(a6),d0        ;; ウィンドウレコードへのポインタ
446:         beq     _TmRet                  ;; ウィンドウがない
447: *
448: * ウィンドウを廃棄する
449: *
450:         move.l  d0,-(sp)
451:         .dc.w   __WMDispose             ;; ウィンドウを廃棄
452:         addq.l  #4,sp
453: *
454: * <tEdit> レコードを廃棄する
455: *
456:         move.l  _TmTEHdl(a6),-(sp)
457:         .dc.w   __TMDispose
458:         addq.l  #4,sp
459: *
460: * 終了する
461: *
462: _TmRet:
463:         lea     _TmGotTxt(pc),a0
464:         move.l  _TmgotLen(pc),d0
465: _TmRet2:
466:         movem.l (sp)+,d1-d7/a1-a2
467:         unlk    a6
468:         rts
```

▶１年がバレンタインデーに始まり，バレンタインデーに終わるはずの光君の活躍が２月号ではなかった。不思議だ。
高麗　道也(19)徳島県

リスト8 フォントパターンの配置

```
  1: * ファイル名:getstr.s
  2: *
  3: ************************************************
  4: *
  5: * これは汎用の1行入力ウインドウを実現する
  6: *
  7: ************************************************
  8: _TmWinPtr       equ     -4
  9: _TmTEHdl        equ     -8
 10: _TmActive       equ     -10
 11: _TmBound        equ     -18
 12: _TmEvMsk        equ     -20
 13: _TmEvRec        equ     -38
 14: _TmParam        equ     -40
 15:                 .text
 16:
 17: .offset 0
 18: teDestRect:     ds.w    4       ;; 内部で使用
 19: teViewRect:     ds.w    4       ;; ビューレクタングル
 20: teDandVRect:    ds.w    4       ;; 内部で使用
 21: teHText:        ds.l    1       ;; 編集テキストへのハンドル
 22: teLenMax:       ds.l    1       ;; 編集テキストの入力最大数
 23: teLength:       ds.l    1       ;; 既に入力されている編集テキストの長さ
 24: teRsv0:         ds.l    1       ;; 内部で使用
 25: teSelStart:     ds.l    1       ;; セレクト範囲の開始位置
 26: teSelEnd:       ds.l    1       ;; セレクト範囲の終了位置
 27: teSelLine:      ds.l    1       ;; 現在のセレクト行位置
 28: teSeloffSet:    ds.l    1       ;; 現在のバッファのセレクト位置
 29: teRsv1:         ds.l    1       ;; 内部で使用
 30: teLineHeight:   ds.w    1       ;; 改行幅
 31: teTabSize:      ds.w    1       ;; TABサイズ(ドット単位)
 32: teJust:         ds.w    1       ;; 行揃え(0:左寄せ 1:中央寄せ -1:右寄せ)
 33: teRsv2:         ds.w    5       ;; 内部で使用
 34: teInPort:       ds.l    1       ;; <graph> へのポインタ
 35: teCaretTime:    ds.l    1       ;; 内部で使用
 36: teCaretState:   ds.w    1       ;; 内部で使用
 37: teRsv3:         ds.l    1       ;; 内部で使用
 38: teRsv4:         ds.l    1       ;; 内部で使用
 39: teRsv5:         ds.l    1       ;; 内部で使用
 40: teRsv6:         ds.l    1       ;; 内部で使用
 41: teRsv7:         ds.l    1       ;; 内部で使用
 42: teNLines:       ds.l    1       ;; テキストの行数
 43: teLineStarts:   ds.l    1       ;; 内部で使用
 44: tEdit
 45:         .text
 46:
 47:         .even
 48: _TmWinTitle:
 49:         .dc.b   14,'なにか入力して'
 50:
 51:         .even
 52: _TmdRect:
 53:         .dc.w   6*2,4,6*256,20
 54: _TmvRect:
 55:         .dc.w   6*2,4,300-6*2,20
 56:
 57: _TmGotTxt:
 58:         .ds.b   512
 59: _TmgotLen:
 60:         .dc.l   0
 61:
 62:         .even
 63: _TmGetStr:
 64:         link    a6,#_TmParam
 65:         movem.l d1-d7/a1-a2,-(sp)       ;; レジスタを保存
 66: *
 67:         clr.l   _TmWinPtr(a6)           ;; ウィンドウレコードへのポインタ
 68:         clr.l   _TmTEHdl(a6)            ;; <tEdit> のハンドルをクリアする
 69:         clr.w   _TmActive(a6)           ;; アクティブの時にオン
 70: *
 71: * ウィンドウの位置と大きさを計算する
 72: *
 73:         .dc.w   __TSGetWindowPos        ;; ウインドウ開始位置
 74:         move.l  d0,_TmBound(a6)
 75:         add.l   #$012c0014,d0           ;; ウインドウの大きさ
 76:         move.l  d0,_TmBound+4(a6)
 77: *
 78: * 自分のタスクIDをウィンドウに登録する
 79: *
 80:         .dc.w   __TSGetID               ;; 自分のタスクIDを得る
 81: *
 82: * ウィンドウをオープンする
 83: *
 84:         move.l  d0,-(sp)                ;; タスクID
 85:         move.w  #$ffff,-(sp)            ;; クローズボックスあり
 86:         move.l  #$ffffffff,-(sp)        ;; 一番手前
 87:         move.w  #$200,-(sp)             ;; 定義関数のID
 88:         move.w  #$ffff,-(sp)            ;; ウィンドウを表示する
 89:         pea     _TmWinTitle(pc)         ;; タイトル
 90:         pea     _TmBound(a6)            ;; ウィンドウサイズ
 91:         clr.l   -(sp)                   ;; メモリを確保する
 92:         .dc.w   __WMOpen
 93:         lea     26(sp),sp
 94:         tst.l   d0
 95:         bmi     err_Open                ;; オープンできなかった
 96:         move.l  a0,_TmWinPtr(a6)        ;; ウィンドウレコードへのポインタをセーブ
 97:         pea     (a0)
 98:         .dc.w   __GMSetGraph            * <current graph> にセット
 99:         addq.l  #4,sp
100: *
101: * アクセスビットをセット
102: *
103:         move.w  #%0011,-(sp)
104:         .dc.w   __GMAPage
105:         addq.l  #2,sp
106: *
107: * フォアグランドカラーをセット
108: *
109:         move.w  #%1011,-(sp)
110:         .dc.w   __GMForeColor
111:         addq.l  #2,sp
112: *
113: * バックグランドカラーをセット
114: *
115:         move.w  #%1001,-(sp)
116:         .dc.w   __GMBackColor
117:         addq.l  #2,sp
118: *
119: * 文字の属性をセット
120: *
121:         move.w  #0,-(sp)                        ;; 6
122:         .dc.w   __GMFontKind
123:         addq.l  #2,sp
124: *
125: * テキストエディトのレコードを作成する
126: *
127:         move.l  _TmWinPtr(a6),-(sp)
128:         pea     _TmvRect(pc)                    ;; view
129:         pea     _TmdRect(pc)                    ;; dest
130:         .dc.w   __TMNew2
131:         lea     12(sp),sp
132:         bmi     err__TmTEHdl                    ;; エラー
133:         move.l  a0,_TmTEHdl(a6)                 ;; ハンドルをセーブする
134:         move.l  (a0),a0
135:         move.l  #256+88,teLenMax(a0)    * 最大文字数
136:         move.w  #12,teLineHeight(a0)    * 改行幅
137: *
138: * 該当するイベントのタイプを指定する
139: *
140:         move.w  #%0011_0010_1000_0010,_TmEvMsk(a6)
141: *
142: * イベント処理ループ
143: *
144: _TmEvLoop:
145:         pea     _TmEvRec(a6)                    ;; イベントレコード
146:         move.w  _TmEvMsk(a6),-(sp)              ;; イベントマスク
147:         .dc.w   __TSEventAvail                  ;; タスクを切り替える
148:         addq.l  #6,sp
149: *
150: * イベント発生時の処理を呼び出す
151: *
152:         move.w  _TmEvRec+0(a6),d0               ;; 発生したイベント
153:         and.w   #$000f,d0
154:         beq     _TmEvNull
155:         cmpi.b  #1,d0
156:         beq     _TmEvMSLDWN
157:         cmpi.b  #3,d0
158:         beq     _TmEvMSRDWN
159:         cmpi.b  #5,d0
160:         beq     _TmEvKey
161:         cmpi.b  #7,d0
162:         beq     _TmEvUpdate
163:         cmpi.b  #9,d0
164:         beq     _TmEv_TmActive
165:         and.b   #$fe,d0
166:         cmpi.b  #12,d0
167:         beq     _TmEvSys12
168:         bra     _TmEvLoop
169: **********************************************
170: *
171: * Null Event Processing
172: *
173: _TmEvNull:
174:         move.l  _TmWinPtr(a6),-(sp)
175:         .dc.w   __GMSetGraph
176:         addq.l  #4,sp
177: *
178: * テキストマンを呼び出す
179: *
180:         pea     _TmEvRec(a6)
181:         move.l  _TmTEHdl(a6),-(sp)
182:         .dc.w   __TMEvent
183:         addq.l  #8,sp
184:         bra     _TmEvLoop
185: *******************************************
186: *
187: * Mouse Left Button Down Processing
188: *
189: _TmEvMSLDWN:
190: *
191: * 自分のウィンドウかを調べる
192: *
193:         move.l  _TmWinPtr(a6),a2
194:         cmp.l   _TmEvRec+2(a6),a2
195:         bne     _TmEvLoop                       ;; 他のウィンドウだった
196:         tst.w   _TmActive(a6)
197:         bne     _TmMslCallWM                    ;; 前からアクティブだった
198: *
199: * ウィンドウを切り替える
200: *
201:         pea     (a2)
202:         .dc.w   __WMSelect                      ;; ウィンドウを切り替える
203:         addq.l  #4,sp
204:         move.l  _TmEvRec+10(a6),-(sp)
205:         .dc.w   __WMFind                        ;; マウスの位置
206:         addq.l  #4,sp
207:         cmp.w   #inContent,d0
208:         beq     _TmMslCallTEM                   ;; コンテントリージョンの中
209:         cmp.w   #inDrag,d0
210:         bne     _TmMslNextEv                    ;; タイトルバーの上ではない
211:         .dc.w   __EMLStill
212:         beq     _TmMslNextEv                    ;; マウスがはなされた
213: *
214: * ウィンドウマンを呼び出す
215: *
216: _TmMslCallWM:
217:         pea     _TmEvRec(a6)
218:         pea     (a2)                            ;; ウィンドウレコードへのポインタ
219:         .dc.w   __SXCallWindM
220:         addq.l  #8,sp
221:         cmp.w   #inClose,d0                     ;; クローズボックスが押されたら
222:         beq     _TmRetKey                       ;; 終了する
223: *
224: * テキストマンを呼び出す
225: *
226: _TmMslCallTEM:
227:         move.l  _TmWinPtr(a6),-(sp)
228:         .dc.w   __GMSetGraph                    * <current graph> にセット
229:         addq.l  #4,sp
230: *
231: * <tEdit> レコードのビューレクタングルに入っているかを調べる
232: *
```

▶今年の目標「陰で悪いことをしていても，胸を張って生きる」　中村　健(21)埼玉県

```
233:         .dc.w    __EMMSLoc                ;; マウスのローカル座標を得る
234:         move.l   d0,-(sp)
235:         move.l   _TmTEHdl(a6),a0
236:         move.l   (a0),a0
237:         pea      teViewRect(a0)           ;; ビューレクタングル
238:         .dc.w    __GMPtInRect
239:         addq.l   #8,sp
240:         beq      _TmMslNextEv
241:         pea      _TmEvRec(a6)
242:         move.l   _TmTEHdl(a6),-(sp)
243:         .dc.w    __TMEvent
244:         addq.l   #8,sp
245: *
246: * イベントレコードを取りのぞく
247: *
248: _TmMslNextEv:
249:         pea      _TmEvRec(a6)             ;; イベントレコード
250:         move.w   _TmEvMsk(a6),-(sp)       ;; イベントマスク
251:         .dc.w    __TSGetEvent             ;; イベントを除く
252:         addq.l   #6,sp
253:         bra      _TmEvLoop
254: *******************************************
255: *
256: * Mouse Right Button Down Processing
257: *
258: _TmEvMSRDWN:
259: *
260: * 自分のウィンドウかを調べる
261: *
262:         move.l   _TmEvRec+2(a6),d0
263:         cmp.l    _TmWinPtr(a6),d0
264:         bne      _TmEvLoop                ;; 他のウィンドウだった
265: *
266: * イベントを取り除く
267: *
268:         pea      _TmEvRec(a6)             ;; イベントレコード
269:         move.w   _TmEvMsk(a6),-(sp)       ;; イベントマスク
270:         .dc.w    __TSGetEvent
271:         addq.l   #6,sp
272: *
273: * <current graph> にセット
274: *
275:         move.l   _TmWinPtr(a6),-(sp)
276:         .dc.w    __GMSetGraph
277:         addq.l   #4,sp
278: *
279: * テキストマンを呼び出す
280: *
281:         pea      _TmEvRec(a6)
282:         move.l   _TmTEHdl(a6),-(sp)
283:         .dc.w    __TMEvent
284:         addq.l   #8,sp
285:         bra      _TmEvLoop
286: *******************************************
287: *
288: * Key Down Processing
289: *
290: _TmEvKey:
291:         tst.w    _TmActive(a6)
292:         beq      _TmEvLoop                ;; アクティブではない
293: *
294: * イベントを取り除く
295: *
296:         pea      _TmEvRec(a6)             ;; イベントレコード
297:         move.w   _TmEvMsk(a6),-(sp)       ;; イベントマスク
298:         .dc.w    __TSGetEvent
299:         addq.l   #6,sp
300:         cmp.w    #$0d,_TmEvRec+4(a6)
301:         bne      keyProc
302:
303: *******************************************
304: * リターンキーの処理
305: *******************************************
306: *
307: * テキストを得る
308: *
309: _TmRetKey:
310:         move.l   #512,-(sp)
311:         pea      _TmGotTxt(pc)
312:         move.l   _TmTEHdl(a6),-(sp)
313:         .dc.w    __TMGetText
314:         lea      12(sp),sp
315:         lea      _TmgotLen(pc),a0
316:         move.l   d0,(a0)
317:         bra      _TmEndPr
318:
319: *******************************************
320: * キー入力処理
321: *******************************************
322: keyProc:
323: *
324: * <current graph> にセット
325: *
326:         move.l   _TmWinPtr(a6),-(sp)
327:         .dc.w    __GMSetGraph
328:         addq.l   #4,sp
329: *
330: * テキストマンを呼び出す
331: *
332:         pea      _TmEvRec(a6)
333:         move.l   _TmTEHdl(a6),-(sp)
334:         .dc.w    __TMEvent
335:         addq.l   #8,sp
336:         bra      _TmEvLoop
337: *********************************************
338: *
339: * Update Processing
340: *
341: _TmMasterUpd:
342:         bsr      EV_UPDATE
343:         bra      _TmEvLoop
344:
345: _TmEvUpdate:
346: *
347: * 自分のウィンドウかを調べる
348: *
349:         move.l   _TmEvRec+2(a6),d0
350:         cmp.l    wPointer(a5),d0
351:         beq      _TmMasterUpd
352:         cmp.l    _TmWinPtr(a6),d0         ;; ウィンドウレコードへのポインタ
353:         bne      _TmEvLoop                ;; 自分のウィンドウじゃない
354: *
355: * <current graph> にセット
356: *
357:         move.l   _TmWinPtr(a6),-(sp)
358:         .dc.w    __GMSetGraph
359:         addq.l   #4,sp
360: *
361: * カーソルを消す (WMUpdate() をコールする前に必ずカーソルを消す)
362: *
363:         clr.w    -(sp)
364:         move.l   _TmTEHdl(a6),-(sp)       ;; ハンドル
365:         .dc.w    __TMCaret
366:         addq.l   #6,sp
367: *
368: * クリップリージョンをセット
369: *
370:         move.l   _TmWinPtr(a6),-(sp)
371:         .dc.w    __WMUpdate
372: *
373: * 画面を再表示する
374: *
375: *
376: * <current graph> にセット
377: *
378:         move.l   _TmWinPtr(a6),-(sp)
379:         .dc.w    __GMSetGraph
380:         addq.l   #4,sp
381: *
382: * テキストを表示する
383: *
384:         pea      _TmvRect(pc)             ;; 囲む四角形
385:         move.l   _TmTEHdl(a6),-(sp)       ;; <tEdit> レコードへのハンドル
386:         .dc.w    __TMUpDate
387:         addq.l   #8,sp
388: *
389: * クリップリージョンを戻す
390: *
391:         .dc.w    __WMUpdtOver
392:         addq.l   #4,sp
393:         bra      _TmEvLoop
394: ***********************************************
395: *
396: * Activate Processing
397: *
398: _TmEv_TmActive:
399: *
400: * 自分のウィンドウかを調べる
401: *
402:         move.l   _TmEvRec+2(a6),d0
403:         beq      _TmEvLoop                ;; どのウィンドウにも該当しない
404:         cmp.l    _TmWinPtr(a6),d0         ;; ウィンドウレコードへのポインタ
405:         bne      inact                    ;; 自分のウィンドウじゃない
406: *
407: * 自分のウィンドウがアクティブになった
408: *
409:         move.w   #1,_TmActive(a6)         ;; アクティブフラグをセット
410: *
411: * 該当するイベントのタイプを指定する
412: *
413:         move.w   #%0011_0010_1010_1011,_TmEvMsk(a6)
414:         bra      _TmEvLoop
415: *
416: * 他のウィンドウがアクティブになった
417: *
418: inact:
419:         clr.w    _TmActive(a6)
420: *
421: * 該当するイベントのタイプを指定する
422: *
423:         move.w   #%0011_0010_1000_0010,_TmEvMsk(a6)
424:         bra      _TmEvLoop
425: *******************************************
426: *
427: * System Event Processing
428: *
429: _TmEvSys12:
430:         bra      _TmEvLoop
431: *******************************************
432: err_Open:
433:         moveq.l  #-1,d0
434:         bra      _TmRet2
435:
436: err__TmTEHdl:
437:         move.l   _TmWinPtr(a6),d0         ;; ウィンドウレコードへのポインタ
438:         beq      _TmRet                   ;; ウィンドウがない
439:         move.l   d0,-(sp)
440:         .dc.w    __WMDispose              ;; ウィンドウを廃棄
441:         addq.l   #4,sp
442:         bra      _TmRet2
443:
444: _TmEndPr:
445:         move.l   _TmWinPtr(a6),d0         ;; ウィンドウレコードへのポインタ
446:         beq      _TmRet                   ;; ウィンドウがない
447: *
448: * ウィンドウを廃棄する
449: *
450:         move.l   d0,-(sp)
451:         .dc.w    __WMDispose              ;; ウィンドウを廃棄
452:         addq.l   #4,sp
453: *
454: * <tEdit> レコードを廃棄する
455: *
456:         move.l   _TmTEHdl(a6),-(sp)
457:         .dc.w    __TMDispose
458:         addq.l   #4,sp
459: *
460: * 終了する
461: *
462: _TmRet:
463:         lea      _TmGotTxt(pc),a0
464:         move.l   _TmgotLen(pc),d0
465: _TmRet2:
466:         movem.l  (sp)+,d1-d7/a1-a2
467:         unlk     a6
468:         rts
```

▶３月号はMIDI特集ですか。恒例行事となった感がありますね。私もようやく購入しまして財布が真っ赤（？）になっています。これでしばらくゲームも買えそうにありません。しばらくはこっちのキーボードを練習します。　臼渕　啓明(24)神奈川県

私の作品制作

プロジェクトチーム DōGA かまた ゆたか

今回はCGAコンテストも間近に迫ったことですので，予定を変更して，入賞決定者の方にCGA制作上の苦労話を語っていただくことにしました。テクニックやイメージ作り，モデリングなどの点で，きっと皆さんの参考になることと思います。

CGAコンテスト特別企画第1弾ということで，応募作品の制作者の方数名に，お話をうかがってみました。

はじめに

先日ある友人Nが，新しい作画アルゴリズムを思いついたといってきた。

N：メタボールの一種なんやけど，メタボールって等電位面を球としていたために，計算量が膨大になっているやん。だから，立方体を等電位面とすれば高速になるはずや。

か：なかなか面白いアイデアやけど，メタボールは，球と球とを近づけていくと融合して，ヒョウタンのような形になるんやんか。立方体同士を近づけるとどう融合するんや？

N：そんなん考えてへん。でも，命名だけはしてん。……メタトウフ。

か：おまえ名前だけで遊んどるな！ ……なら，おれも思いついたぞ。球と球を接近させると，その距離に反比例した数のワイヤーフレームでつなげるんや。

N：？

か：……メタナットウ。

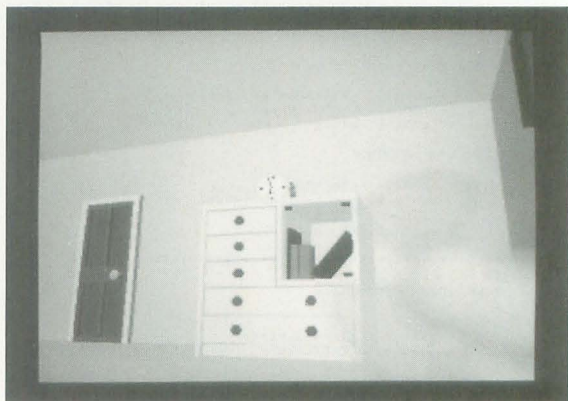
＊

おひさしぶりのかまたです。昨年9月号以来の登場ですね。前回，前々回とMAX田口君に任せていましたが，いい加減なことを書いて皆さんにご迷惑をかけるなんてことはなかったでしょうか（まぁ私も大差ないけど）。

さて，いよいよ「CGAコンテスト」が近づいてきました。実は，昨年12月の中ごろまで作品の応募が1点もなく，どうなっているんだとメチャメチャあせったのですが，年末間際になって，続々と到着しました。皆さんもギリギリまでがんばったのですね。その努力が実って，はっきりいってなかなか見応えのある作品が集まっています。やはり制作者の方々も，昨年の作品を研究していらっしゃるのでしょう。

ということで今回は，一次審査が終了して入選が決定した方にお話をうかがってみました。皆さんの作品制作や，次回のコンテストを目指す参考になることでしょう。

インタビュー

●KMC［京大マイコンクラブ］

KMCは，DōGA CGAシステムの開発だけでなく，作品制作も活発なクラブで，CGAコンテストにも第1回から，質の高い作品を発表し続けている。今回はなんと一度に4作品を制作し，すべて入選を果たした。もちろんこの4作品はすべて制作者が異なる。人材，機材ともに豊富なクラブだからこそなせる技といえよう。

今回は，4作品の中でも本命と評判の高い「CLOCK」の監督，上原哲太郎さんに話をうかがった。「CLOCK」は，壁時計が，部屋の向い側の壁に掛かっている振子時計に会いに行くための，たった10メートル足らずの苦難の道のりをコミカルに描いた作品である。

＊

か：それではまず，「CLOCK」の見どころを解説していただけないでしょうか。

上：う～ん，技術ですね。CGAとしての技術。我々KMCは，CGAシステムの開発も行っている立場ですから，このシステムでこんなこともできるんだぞということを，世の中に示さないといけないと思うんです。

か：具体的にいうと？

上：まず，時計の文字盤なんかのマッピングです。今年はマッピングができるから，それを利用した作品をやろうというところからこの作品が生まれたのです。それから，去年の「レイズピー」で使用された“ぼかし”を応

用して，ピントのずれによる遠近法（25ページ参照）をやってみました。

か：作品中に半透明のカーテンが風でなびくシーンがありましたが，あれも通常のやり方では表現できないのではないですか？

上：ええ，専用のプログラムを開発しました。ほかに特殊なプログラムといえば，512×512の解像度のアニメーションも数カット行っています。ただ，512×512の画像もビデオに収録すると，256のアンチエイリアスの画面とほとんど区別がつきませんでした。斜めの線が妙になめらかに見えるカットが512です。

か：今年も入選作品集のビデオを出しますから，その辺はひとコマずつ丁寧に見てもらいましょう。

上：それから新しい試みとしては，すべての効果音をPCMでしました。

か：なるほど，それで画面と効果音が完全に一致するんだ。扇風機のアップのときにはモーターの音が大きくて，ぱっとロングに切り替わった瞬間音も小さくなったから，これはまた気合いの入ったアフレコをしているなと感心していたんですよ。

上：でも1チャンネルしかないから，音が重なったところは全部重なったPCMを用意しなくちゃいけないんですよ。曲もPCMです。

か：技術だけでなく，ストーリー性など，作品としての完成度も非常に高かったと思いますが。SIGGRAPHで発表されるPIXER社の作品を連想させるものがありますね。

上：やはり，「Luxo Jr.」の路線は意識しました。前半のストーリーは，20分ほどで出来上がってしまいましたが，後半は悩みました。今でも悩んでいます。

か：振子時計に会うための数々の障害をどうクリアするのか，本当に辿りつけるのか。なかなか楽しめました。

上：そのあたりが作品としての見どころといえるでしょう。「Luxo Jr.」でもそうですが，CGAシステムは生物を描くのに限界があるので，生き物でないものを動かしました。形状データは極力簡単なものにすることで，動きに凝ることができました。

か：個人的な意見ですが，この「CLOCK」はKMCの歴代の作品の中でも群を抜いていると思うのですが。

上：そうだとすれば，これまでの作品がほとんど個人作品だったのに対して，「CLOCK」では団体作業が行われたという点が大きかったのでしょう。各個人の欠点もお互いにカバーできましたから。

か：団体作業でやると，どういう点で有利ですか？

上：実は，私はフレームソースをひとつたりとも記述していないんです。フレームソースのことなんかよくわかってないから，平気で無茶をいえるんです（笑）。だから，こだわらないといけない，自分でもこだわりたいところには，徹底して妥協を許さなかったんですよ。

か：それは，団体作業のメリットというより，単に上原さんが楽をしたということでしょ（笑）。

上：私もちゃんとつき合って徹夜してますよ（笑）。私が，とにかく“こういうカットなんだ，こんなふうに動くんだ”と，ジェスチャーや，本当の時計を引っ張り出して実演したりしてみんなに説明するんですよ。“このとおりに作ってくれ”っていって作らせて，自分のイメージと同じものができるまで，“あとちょっと左だ，もうちょっと早く”とかいって，何度も繰り返しさせたんです。

か：極悪～（笑）。そうすると，作業の大部分がフレームソースの制作だったと思いますが，何人で行われたのですか？

上：3，4人ってとこでしょう。

か：その方法で，上原さんがひとりで行われたときより，3，4倍の効率が実現できたのですか？

上：そうですね。ひとりに説明して，それが終わったら次のヤツに説明してという作業を続けていくわけです。

か：なるほど。こんなイメージっていうだけで，ちゃんと映像ができる。それって，究極のマンマシンインタフェイスっていいません（笑）？　それでは最後に，KMCのほかの作品についてお願いします。

上：「connection」は，3日間で完成させたという，アイデアとセンスの作品です。“前衛”を味わってください。「デスペラード（予告編）」の原作は，KMC内で書

CGAコンテスト事務局より

「第3回 アマチュアCGAコンテスト 入賞作品上映会」のお知らせ

CGAコンテストとは……なんて，もう説明しないぞ。とにかく，CGアニメーションに興味があるヤツは，みんなで見にいけばいいんだ。見逃したら，一生後悔するぞ。まぁ，作品集のビデオテープも出るから，そっちを見てもいいけどな。でもやっぱり，ほかのみんなより早く見たいってもんじゃないか。それに，ビデオにはないいろいろな余興もあるというウワサだしな。うんうん，やっぱり見にいかなきゃ男じゃないね。女の子でもいいけど……。

東京地区
日時　1991年3月2日（土曜）　PM1：30 開場　PM2：00 開演　～5：00
場所　YAMAHAホール　東京都中央区銀座7－9－14
　　　地下鉄銀座駅 A3出口 南へ400m

大阪地区
日時　1991年3月17日（日曜）　AM11：00～（随時）
場所　J＆Pテクノランド イベントホール
　　　地下鉄堺筋線 恵比寿町駅真上

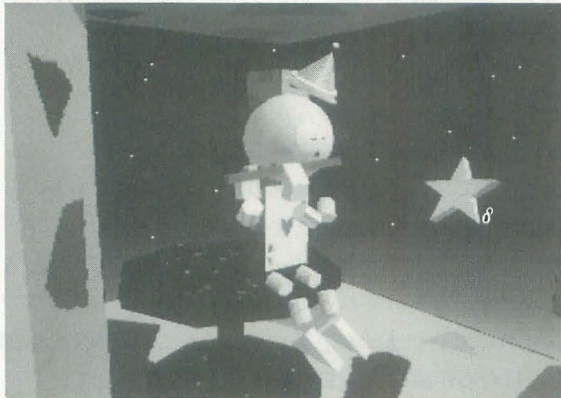
注意がいくつかあるから，よく読めよ。まず，場所も日時も去年とは違うから気をつけるんだぜ。去年やった市ヶ谷のシャープのホールへ行ったって，ネクタイ締めたおっちゃんが，会議をしてるぞ。会議のじゃまはいかんな。

YAMAHAホールは，1階がヤマハのお店になっていたな。ホールは5階だったっけな。外からは目立たないが，中は立派なホールだ。驚くんじゃないぞ（本当は驚いてほしいけど）。

入場は無料だ。ただ，YAMAHAホールは，会場定員が500名なんで，先着順だぞ。満員のときは，会場で売っているビデオを手に入れるだけで，勘弁してくれ。すまん。

かれたSF巨編で，テーマ曲，ゲームまでできていますが，ちゃんと映像化すると，映画数本分になってしまいます。ということで，予告編です。本編はありません。予告編というジャンルの作品なのです。最後に「ゲッピーロボ」は，おなじみ“ピー”シリーズです。このゲッピーには，厚紙で試作された高さ40cmの模型があります。この模型，作品の中で行う変形合体を，実際に行うことができるのがポイントです。

か：今後も，楽しい，KMCらしい作品を作っていってください。

●寺尾　響子さん

主婦&イラストレーター。今回のCGAコンテスト唯一の女性の入選者である。昨年から，サイクロンなどを用いて，仕事にCGも取り入れ始めた。CGAシステムを入手してまだ3カ月で，アニメーション作品への挑戦は初めて。

作品「HEART」は，トランプのハートのエースがひとりで笛を吹く様子をかわいいタッチで描いている。

＊

か：私が「HEART」を拝見させていただいて最初に気がついたのは，カメラ(視点)を固定させた，いわゆるフィックスのカットで構成されているという点なんですが。

寺：この作品を作り始めたのが12月に入ってからなので，時間的に余裕がなかったという面もあるのですが，初めて作品を見る方にとっては，視点を動かしすぎたり，カットがパッパと切り替わるのは見にくいと思ったからです。

か：フィックスは映像の基本ですからね。ただそれらのカットが，静止画としてもみんなちゃんと絵になっているので，さすがにイラストをやっている方だなと感心しました。

寺：それは意識して制作しました。でもそれって，かまたさんがOh!Xの連載で書いていらしてましたよね。だから，私忠実に守ったんですけど（笑)。

か：（そういえば，その話は前に掲載したかなと思いつつ）そうでしたね……。そういえば，マッピングを多用されていますね（急に話題を変える)。

寺：CGの基本的なテクニックとして，物体数を増やさないで，表現を豊かにするためにマッピングは有効ですから。

か：（サイクロンなどのレイトレーシングソフトでは確かにそのとおりだが，CGAシステムのマッピングは，高速化のために手抜きをしているから，あまりマッピングを多用するのは危険ではないのかなと思いつつ）そのとおりですね……。

読者通達事項

[タケルによるサービス開始のお知らせ]

“タケル”ってご存じですか？ J&Pのような量販店などに設置してあるソフトの自動販売機です。ブラザー工業のご好意により，当チームでもこのタケルを利用した各種サービスを行うこととなりました。

なにしろ自動販売機ですから，パソコン通信ができない皆さんでも，好きなときに，その場で，好きなものを入手できますし，当方の作業負担も軽減されます。

当チームは，ご存じのように非営利団体ですので，基本的に無料で提供していきます（例によって，カンパは受け付けます）が，各利用者には，タケル使用料として1,000円〜1,500円程度が必要になります。当チームから送ってもらっても，ネットからダウンロードしても，それなりのお金はかかるのですから，ご了承ください。

タケルが設置してあるお店は，各都道府県に最低1店はあるそうで，今年も設置店を大幅に増やすそうです。

〈2月下旬より〉

●バージョンアップサービス
●CGアニメーション体験システム
●サンプルデータ集

〈5月中旬より〉（予定）

●KO-WINDOW

Oh!X 1月号のGraphic Galleryにも掲載されましたオリジナルウィンドウシステムです。ウィンドウシステムに触れてみたいという方やウィンドウ上のアプリケーションを開発してみたいという方など，かなりのパワーユーザー向き。ソースもすべてついています。

●そのほか新しいツールなど

[バグレポート]

・AMAP

AMAPによって，マッピング用の形状データを生成したあと，RENDで作画しようとすると，ごくまれに“数値が必要です”とかいうエラーメッセージが大量に表示される現象が確認されました。

これは，AMAPで変換した際，“uvpoly”にしなければいけないところを“shade”にしているからです（uvpolyの数値は5つに対して，shadeは6つなので，数値が必要というエラーになる)。

対応策は簡単で，そのような現象が現れた形状データをED(エディタ)で読み込み，“shade”をすべて“uvpoly”に置換してやるだけで，正常なデータとなります。

・マッピング（REND）

RENDのマッピング機能はあくまでもおまけであり，高速化のためにいろいろ手抜きがされています。

たとえば，道路にセンターラインをマッピングするようなケースでは，センターラインの間隔が遠くほど小さく，手前ほど大きくなるはずですが，すべて等間隔になってしまいます。これは，3D/2D変換を行ったあと，つまり3Dの情報が失われたあとに絵を張りつけているからです。

しかし，この現象は意外と目立たず，いわれてみないと気がつかないものです。それに，道路を1枚の面にせず，いくつかに分割することで，回避できます。

さらにやっかいなのが，ビルの壁に窓をマッピングするようなケースです。視点と壁の距離が小さく，視線と壁が平行に近いとき，窓が大きくゆがんだり，引き延ばされたりします。

“こんなのバグだ”とRENDの作者であるレイバー小林に詰めよったのですが，小林君は難しい数式や私にわからないようなアルゴリズムで反論し，結局高速性を維持するためにやむをえない仕様であるということになってしまいました。小林強し！

今月の格言　　取れないバクは仕様

寺：特にアニメーションの場合，物体数が増えると，動きをつけるのがたいへんでしょうから，物体はできるだけシンプルにしようと思ったわけです。そのかわり，何度もレンダリングして，アニメーションをして，というようにシミュレーションを行おうと。

か：（その通りだと思いつつ）その通りですね。

寺：物体に凝りだすと，作品制作の比重が形状デザインのほうばかりにいって，木を見て森を見ずってことになるんじゃないですか。それよりは，作品の流れや，作者が訴えたいイメージを追求するほうが大切だと思うんです。

［この間私は，寺尾さんお手製のなんか難しい名前のついたお菓子をいただいている。おいしかった］

か：寺尾さんの場合，初めての作品でもあり，CGAシステムを使いだして間もないということで，いろいろ困ったこともあったのではないですか。

寺：そうですね。構造体は使用しませんでした。構造体でやれば，もっと高度な動きができるのはわかっていたのですが，なにしろ時間がなかったので，すべてFFEでデザインしました。

か：年末でお忙しかったでしょう。

寺：そうなんですよ。一応主婦ですから，お節料理も作らないといけませんし。ですから，削れるところはどんどん削って，とにかく完成することを目指しました。

か：やっぱり作品は，出来上がることがいちばん大切ですね。

寺：これから，初めて作品を制作される方は，自分の今の力の70％ぐらいでできる作品を目指すのがいいと思います。つまり，まだあまり使いこなしていない機能や，新しいツールを使わずにできる作品ということです。それから，作り始める前に，その作品のイメージや，何を作りたいかということを明確にしておくのが大切だと思います。作品を作りながらイメージをだんだん広げていくっていう方もいらっしゃるとは思いますが，初心者の場合，途中で迷いだすときりがなくなってしまうんです。これは，イラストの仕事をしている経験からいうのですが，限られた時間で作品として仕上げるためには，そういった，制作の過程で出てきた迷いとか，場合によってはアイデアとかも，次の作品のときに生かそうと割り切ってしまうことが必要です。

か：初めて制作されただけあって，説得力のあるアドバイスですね。

寺：ほかには，CMなどの影響で見る方の気が短くなってきてますので，短い作品で勝負したほうがいいと思います。私自身，今回の作品は少し長すぎた感じがします。

か：長いカットは，「HEART」のように，アングルをいろいろ変えるなど，見せ方のテクニックもありますが，基本的に作品は，無理に長くするよりは，短くまとめるべきでしょうね。

寺：音楽でカバーするのもよいですね。みんなが知っている曲とか，のりやすいリズムとか。

か：「HEART」の音楽はどのようにされたのですか。

寺：あれは，ジャズのスタンダードです。同じフレーズの繰り返しが多いので，長さの調節ができます。先に映像を作って，ストップウォッチで計ってから，演奏しました。ミックスダウンまで含めて，曲にかけたのは1時間ぐらいです。

か：BGMの演奏もできるし，イラストレーターのセンスもある。今後もパーソナルCGAをリードする制作活

一太郎古賀の新ツール紹介

はじめまして。DōGAのテクニカルライター（ただのマニュアル書き？）古賀と申します。

さて，当プロジェクトルームでは，さまざまなプログラムが日々開発されています。そこで，このコラムでは新しく開発されたツールを紹介していきたいと思います。

基本的にこのコラムで紹介されたツールは，J＆P HOTLINEのDōGA-SIGにアップロードされます。ただし，新しいツールにはバグがあるのは当たり前ですから，バグ出しの協力のつもりでダウンロードしてください（バグを見つけたらレポートをアップしてね）。また，読者通達事項で紹介しましたように，ソフト自動販売機タケルでこれらのツールが手に入るようになるのは，5月中旬の予定です。

・PFCOMP［PicFile COMPressor］

マッピングやアンチエイリアスなど，調子にのって綺麗な画像を作っていると，画像ファイルがとてつもなく大きくなってしまうことがあります。画像ファイルが25Kバイトを越えると，最新のHANIMを使っても，20フレーム/秒のアニメーションができなくなりますし，ディスクにも多数保存できません。そんなとき，画像データの圧縮効率をよくして，ファイルサイズを小さくしてしまうのが，このPFCOMPです。

アルゴリズムは，非常に似た色が数ドットだけ並んでいるところを同じ色にしているだけです。ですから，アンチエイリアスもかかっていない10Kバイト程度の画像に対してはほとんど効き目はありませんが，25Kバイト程度の画像では，約70％ぐらいに圧縮し，見た目にもほとんど変わりません。

そのほかの機能として，上下をカットした画像ファイルを生成することもできます。ビデオに録画するときは，15kHzで画像が縦方向に伸びるため，画像の上下20ライン程度ははみ出して表示できません。表示できない部分のデータは無駄ですから，カットする（黒で塗りつぶす）ことで，さらにファイルサイズを85％ぐらいに小さくできます。

さらに，下のほうを大きめにカットして字幕を表示する領域を作ったり，上下を大きく切り取り，横長の画面にして「ハイビジョンだ！」とうそぶくこともできます。

〈制作者　ぱーわん八幡（prodige）〉

・WIPER［WIPER］

シーンとシーンの継ぎめの処理には，フェードイン，フェードアウト，オーバーラップなどがありますが，前のシーンの一部が残っている状態で，次のシーンが画面の横や上から挿入されるような表現をワイプといいます（ほんまかいな）。画像ファイルの一部を大きさを変えながら連続的に切り取って，ほかの動画に合成することで，ワイプのような効果を生成するのがWIPERです。

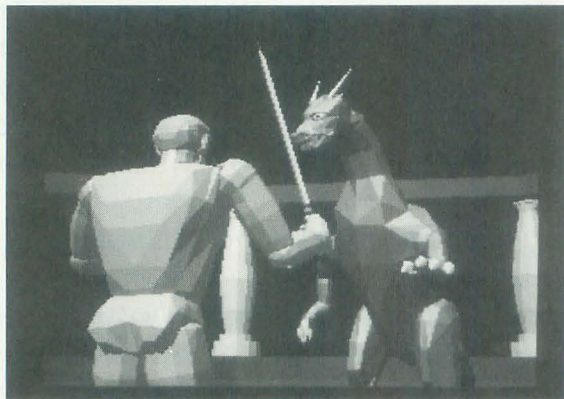
文章で書くとややこしいのですが，このWIPERは結構ユーザーインタフェイスがよいので，見ただけで使い方はだいたいわかります。作品中に何度も用いるものではないのですが，プレゼンテーション色の強いものや，エンディングクレジットなどには効果を発揮するでしょう。

ワイプのパターンとしては，中央から開いたり，窓のブラインドのようなものなどいろいろ用意されています。なかでも面白いのが円形のパターンで，大きく（あるいは小さく）なっていく円を，任意の位置に複数個設定できます。つまり，“トムとジェリー”なんかでよくある，トムの顔とジェリーの顔のところに円が小さくなっていって，“THE END”が表示されるといった表現ができるわけです。

〈制作者　MAJIN中田（prodige）〉

動に期待させていただきます。

寺：CGAは作っていて楽しいものですから，おばあさんになるまで続けていきたいと思います。

●森山　知巳さん

本職は日本画家。Oh!X 1月号のGraphic Galleryをはじめ本連載の教育的指導のコーナーにも何度か出演しているので，ご存じの方も多いだろう。

今回の作品「SWORD」は，Oh!X 1月号で紹介したものの完成版。しかしながら，あの写真からこの作品を想像することは難しい。本編はむちゃくちゃカッコいい。個人的に特にお勧めの1本である。

なお，昨年「ディファイナブル ファンクション」，「超強力宇宙人」で注目を浴びた森山さんとは，別人である。

＊

か：お仕事で描くいかにも日本画という感じの絵と，CGAの「SWORD」は，相反するイメージがあるのですが。

森：仕事では，古典的な手法を取り入れた日本画を描いています。しかし，元来新しいもの好きでもあるので，CGなんかにも挑戦してみたかった。それはあくまで趣味で，仕事の合間に，仕事の反動による制作です。でも，作品を作りだすと，どっちが合間かわからなくなって，家内ににらまれたりしましたが（笑）。

ただの雑談

[そいつはデマだ!]

当チームも地道な（?）活動を続けているうちに，少しずつ知られるようになってきた。すると，本人たちになんの連絡もなしに，雑誌などに掲載されているということが起こってきた。また，ネットなどで，妙なうわさが流れたり，それに尾ヒレがついてとんでもないデマになることもある。今回は，「KO-WINDOW」に関するデマをまとめてみた。

＊

デマ1：KO-WINDOWの制作者は，project team DōGAだ

実は，そうではありません。KO-WINDOWの作者は，レイバー小林です。確かに小林君は，当チームのチーフスタッフですが，このウィンドウシステムは，彼が個人的に趣味で作っただけです。本人は気にしていないようですが，制作者の名前ぐらい正しく紹介してやってください。

デマ2：“KO”は，SX-WINDOWをノックアウトしてやるという意味だ

これがいわゆる根も葉もないデマというやつです。KOを開発していたころは，まだSXが出るということすら知らなかったのですから。では“KO”の意味ですが，単に“小林が作った例のWINDOW”の略です。安直ですいません。

デマ3：KOは，CGAシステムVer.3のインタフェイスのために開発した

繰り返しいいますが，単に小林君の趣味です。ついでに，“Ver.3の開発が活発に行われている”などというデマも打ち消しておきます。Ver.3の開発は，まだ着手されていません。どんなシステムにするかすら決まっていません。KOを使うかどうかも賛否両論があり決まっていません。ですから，年内に発表される可能性はまずないと思っていてください。おあいにくさま。

[from the DARK SIDE]

CGAコンテストや，Oh!XのGraphic Galleryのように，皆さんの目に触れる作品が，いわゆる陽のあたる世界だとすると，パーソナルCGAにはもうひとつ裏の世界がある。それが“DARK SIDE（暗黒面）”!

“DARK SIDE”とは，アニパロを中心とした，著作権に触れる作品の世界である。ダースベイダー氏曰く，“DARK SIDE is POWER!”。ヨーダ曰く“暗黒面に入った者は，二度と戻って来れぬ……”。

最近，水面下でこのDARK SIDEの活動が活発になっている。Z○ンダムの完全変形，グリ○ォンと零式の対決，○トレ○バー2号機。昔掲載したパロレイバーなどのレベルではない（だから掲載もできない）。ご覧いただかないと信じてもらえないだろうが，TVアニメと比較してもなんら遜色ない（いや，超えているかもしれない）。

そんな“DARK SIDE”から“暗黒面のススメ”ともいうべき文書が届いた。

＊

>形状データの作成は恐くない!（みんなで幸せになろうよ）

この手の物体をデザインする場合，まず制作する前に，自分の作りたい物体に関する図面及び模型を入手します。夜毎にそれを眺めながら，どの程度まで詳細に作成するかを考えます（妥協をしてはいけなーい! どうせやるなら徹底的に!）。

ある程度構想がまとまったら，ノギスと定規をおもむろに取り出します。そして，面を構成する部分を自分の納得のいくまで測定します。

あとは，ただディスプレイに向かい，形状データの完成のみを頭において作業を進めるだけです。“このデータは，俺にしかできないのだ!”と自分に思い込ませるのがポイントです。

CADを起動したらまず，作り始める部品（肩とか太股という単位）の輪郭だけを表す3面を描きます。つまり，上面図を拡大して，Z＝0のXY平面に上から見た輪郭を，同様に側面図，正面図に横から見た，前から見た輪郭をそれぞれ描くわけです。あとは，測定しておいたデータや，ノギスを頼りに，1面ずつ張り合わせていきます。

データは部品が完成するごとに，レンダリングを行い，形状の異常を確かめましょう。全部完成したら，全体のバランスを確認しましょう。他人に見せて，おかしいところがあるかどうか意見を聞くとよくわかります（おかしいといわれたら，当然作り直しです）。

CADに早く慣れるために，いちばん最初にものすごく複雑で面数の多いものをデザインするという方法があります。これができれば，次回からは，かなり楽に感じるわけです。

CADは，根性です，センスです。一度身についた考え方と技はそう簡単に変えられません。皆さんもやってみませんか? ほーら，恐いことはないんだよ。恐いのは著作権だけなんだよー。

＊

私は，以前から“アニパロはやめよう”といってきたが，それは“みんながこの路線に走るのは危険だ”という警告であって，このDARK SIDEを“悪”だとか，否定するつもりは毛頭ない。

陽のあたる世界より，はるかにパワフルで，テクニカルで，ハイレベル，マニアック! これを読んでるあなたも，DARK SIDEに引かれるものを感じているはずだ。

この恐ろしい，いや，すばらしい世界を，皆さんにもご覧いただく機会はないものだろうか。CGAコンテストのビデオに入れる……著作権上それは許されない。でもコンテストの発表会という限られた会場でちょっとだけ上映するぐらいなら……。いや，何もいうまい。なぜなら，それが“DARK SIDE”!

か：どういった過程でこの作品は生まれたのでしょうか。
森：もとからグラフィックはやってみたかった。どうせなら3D。さらにアニメーション。動く楽しさがいちばんですからね。ちょうどそのころCGAシステムを手に入れたのです。
か：いつごろですか？
森：一昨年の夏。Oh!Xの連載が始まった直後です。入手したころは，失礼ですが，そんなまとまった作品が作れるようなものだとは思っていなくて，子供を撮影したビデオのオープニングタイトルに使ってみる程度に考えていました。しかし，いざやってみると，わりと簡単にできてしまった。それも面白いものが。ほかの人に見せても，最初のタイトルのところはプロみたいって（笑）。
か：そのころのものがOh!Xでも掲載された「メリーゴーラウンド」ですね。
森：はい。それで，CADでどのくらいの人体モデルができるのだろうと試すつもりであの人体を作ったのです。その時点でも作品としてまとめるつもりはありませんでした。なにしろ，初期のRENDでは，人体ひとつ作画するのに10分以上かかりましたから，途方に暮れていました。でも，コプロを積んで，RENDもバージョンアップされて，3分以内でできるようになった。それじゃ，犬を作って，竜を作ってという具合に作品としてまとめていったのです。
か：はじめてDōGAにあの人体モデルが送られてきたとき，よくやるなぁと感心というかあきれていて，手紙の中に近々竜と格闘しますとありましたが，単なる冗談だと思ってました（笑）。竜などのなめらかな生物的な動きもすごいですね。どのようにされたのですか。
森：Oh!Xに掲載されていた“ラナ君のおたまじゃくし”のしっぽの動かし方を参考にしているだけです。提供されたCGAシステムというツールだけで，特別なことをせずに，どこまでできるのかというのを見せるのが，ある意味でユーザーとしてのお礼ではないでしょうか。プログラムを開発してもらったDōGAに対して。
か：う～ん。特別なことをせずに，これだけの作品ができるってのはちょっと信じがたいですが。でも，アンチエイリアスやスムースシェーディングなどの機能は使っていないのはなぜなのですか？
森：そういうアプローチはしませんでした。時間がなかったというのも事実ですが，アマチュアCGAっていうのは，早いもの勝ちとか，金銭勝負ではないと思ったからです。つまり，はじめてこういう手法を実現したとか，高価なマシンを使えば誰にでもできるようなことを競ってもしかたがないと。
か：私は，アマチュアCGAの新しい技術開発というのは絶対必要で，そういった趣向の作品も重要であると思いますが，確かに単に画質が良い，高価なマシンを用いたというのはくだらないと思います。
森：ええ。NHKの人体の中でも，高価であるだけで，ひどい映像があちこちありました。ああいったのは，アマチュアCGAのアプローチではありません。やはり，制作者が楽しむというか，制作者のバックボーンを生かした作品であってほしいですね。
か：そうですね。今回のコンテストでもそういった作品は多くありましたよ。自動車会社の方が自分の理想の車をデザインして，その30秒CMを制作したとか，NHKの人体の話が出ましたが，医科大学の方が精子と卵子の受精シーンを作られていました。
森：同じ受精シーンでも，そういうのはとってもいいですね。
か：また，この精子の動きが凝っているんですよ。
森：やはり，CGAは動きにメインを置いたほうがアマチュアらしいですね。「SWORD」の制作においても，キャラクターを自由に動かしてみたいという思いがありました。ただ，今こうして見ると，もっと凝りたかった。もっと細部にまでこだわるのが個人の楽しみであり，アマチュアの特権でしょう。バージョンアップサービスがもうちょっと早かったら，また作品も変わってきたんですが。
か：すいません。精進します（笑）。
森：いやいや本当に感謝していますよ。放送局のDVEを使ったような映像作品レベルのものが，時間さえかければ個人でもできるんです。こういった環境が個人の手に入ったというのは，たいへんうれしいし，本当にすばらしいことだと思います。ですから，私にできるお礼としてはこれしかないのですから，今後も作品を作っていきたいと思います。

おわりに

いかがでしたでしょうか。入選者の皆さんは，さまざまな意図やアプローチで作品を制作されているのがおわかりいただけたでしょう。CGAコンテストには，今回ご紹介しました作品に勝るとも劣らない作品が多数よせられています。絶対にお見逃しのないように。

上映会は，東京，大阪以外にも，名古屋，金沢での開催を予定していますが，どうなるかはまだわかりません。それ以外の地域の方のために，今年もビデオを出す予定です。昨年と比較すると，作品の量も増えているので，実費がかさむ可能性がありますが，できるだけ多くの方にご覧いただけるように，努力します。詳しい申し込み方法などは，来月号をご覧ください。

さて次回は，CGAコンテスト特別企画第2弾として，審査員の方々のご意見などをご紹介したいと思います。でも，前回の予告では，「戦えロボット君2」をお届けするといっていたことだし，あてにはなりません（なら，最初から予告をするなって）。

★(で)のショートプロぱーてぃ その18

春のピコピコ

Komura Satoshi 古村 聡

今月はぱーてぃハンズがお休みになってしまいました。しかし，ショートプログラムのほうはゲームが2本と，時計が1本？ ということで，満足していただける内容になっているのではないでしょうか。ハンズは来月のお楽しみに。

illustration : T.Takahashi

ポリポリ。どうもすいませんです。今月のぱーてぃハンズ，お休みになってしまいました。

というのも……，うおーっ！ 時間がないぞ。そうなのです，私はこれでも一応は学生さん（なに，実はこっちが本業なんだろうって？）なので，この時期は試験で大変なのです。ゼイゼイ。明日試験だっていうのに原稿など書いてていいのだろうか（と言いつつ，しっかり現実逃避している私（で）なのであった)。あまりの忙しさに記憶喪失になってしまう。

はっ，私はいったいだれなんだ……？ 私は，私は……，そうか，私は通りがかりの銭形平次さんだったんだあ（なんじゃそりゃ)。

文字で空が落ちてくる

ということで，今月の1本目は埼玉県の渋谷寛さんの作品，「アルファベットの逆襲」です。

アルファベットの逆襲
for X1シリーズ
(CZ-8FB01)
埼玉県 渋谷寛

ひっさしぶりのX1 BASIC用のゲームプログラムですね。遥か空の彼方からアルファベットが攻めてくるのです。地上が彼らに侵略される前にあなたの鮮やかなキーボードさばきで彼らを撃退してください。

アルファベットの逆襲

撃退の方法は降りてくる文字と同じ字をキーボードから打つだけです。地球の存亡は君の指にかかっている！

ああ，久し振りのX1，そのうえ久し振りのピコピコゲームなのであります。そういえばキャラクタしか使っていないゲームプログラムというのはいつ以来でしょうね，本当に（まあ，PCG使ったものはあったわけですけど)。

上からアルファベットが降りてくるのですが，たまに意気地なしの文字がいるのか降りてくる途中で止まってしまったりとか，

リスト1 アルファベットの逆襲

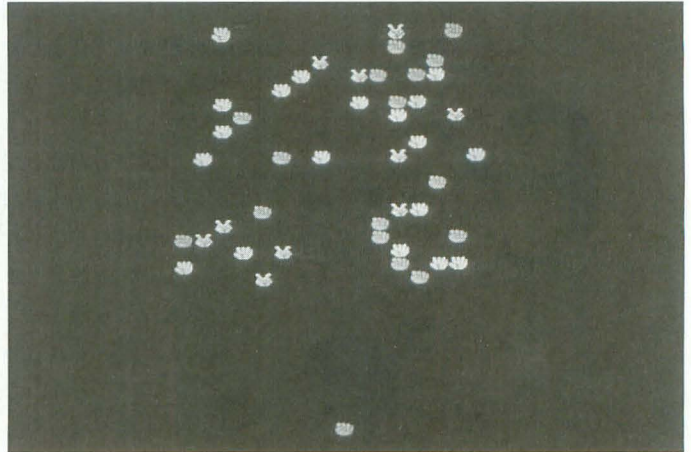
```
1000 DIM CY(26)
1010 GOTO 1500
1020 WIDTH 80
1030 FOR I=0  TO 25
1040    CY(I)=4
1050 NEXT
1060 SCORE=0:SC100=SCORE¥50
1070 KLOOP=25:'TIME
1080 LOCATE 0,3:PRINT"--------------------------------------------------------------------------------"
1090 LOCATE 0,21:PRINT"--------------------------------------------------------------------------------"
1100 LOCATE 25,0:PRINT"REVENGE OF ALPHABETS!!"
1110 LOCATE 10,2:PRINT"SCORE"
1120 LOCATE 15,2:PRINTSCORE
1130 LOCATE 55,2:PRINT"HI-SCORE"
1140 LOCATE 65,2:PRINTHSC
1150 X=INT(RND(1)*26)
1160 L=INT(RND(1)*36)+1
1170 IF SC100<SCORE¥100 THEN SC100=SCORE¥50:KLOOP=KLOOP-5
1180 IF SC100<0 THEN SC100=0
1190 FOR I=1 TO L
1200    LOCATE X*3,CY(X):PRINTCHR$(X+65)
1210    J=0
1220    A$=INKEY$:IF A$<>"" THEN GOSUB 1270
1230    IF J<KLOOP THEN J=J+1:GOTO 1220
1240    CY(X)=CY(X)+1:IF CY(X)>21 THEN 1430
1250 NEXT
1260 GOTO 1150
1270 '
1280 KIN=ASC(A$)
1290    IF KIN<65 THEN RETURN
1300    IF KIN>122 THEN RETURN
1310    IF KIN>90 AND KIN<97 THEN RETURN
1320    IF    KIN>96 THEN KIN=KIN-32
1330 KIN=KIN-65:BEEP:NC=0
1340 FOR I=CY(KIN) TO 4 STEP-1
1350    LOCATE KIN*3,I:PRINT" ";:NC=NC+1
1360          FOR J=0 TO 20:NEXT:'JIKANKASEGI
1370 NEXT
1380 IF NC>0 THEN CY(KIN)=4:SCORE=SCORE+NC
1390 LOCATE 15,2:PRINT SCORE
1400  IF SCORE>HSC THEN HSC=SCORE:LOCATE65,2:PRINTHSC
1410 IF X=KIN THEN GOTO 1150
1420 RETURN
1430 '----GAME OVER---
1440 LOCATE 19,11:PRINT"                                         "
1450 LOCATE 19,12:PRINT"************* GAME OVER *************"
1460 LOCATE 19,13:PRINT"                                         "
1470 LOCATE 19,14:PRINT"              HIT SPACE BAR!             "
1480 LOCATE 19,15:PRINT"                                         "
1490 A$=INKEY$:IF A$<>" " THEN 1490
1500 WIDTH 80
1510 LOCATE18,6:PRINT"****************************************"
1520 LOCATE18,7:PRINT"*                                      *"
1530 LOCATE18,8:PRINT"*          REVENGE OF ALPHABETS        *"
1540 LOCATE18,9:PRINT"*                                      *"
1550 LOCATE18,10:PRINT"*                                      *"
1560 LOCATE18,11:PRINT"****************************************
1570 LOCATE 31,14:PRINT"HIT SPACE BAR!"
1580 A$=INKEY$:IF A$ <> " " THEN 1580
1590 GOTO 1020
```

大時計

ジャンしゅー

代わりにほかの文字が降りてきたりもするのです。そういえばゲームセンターの光線銃のゲームでこういうゲームがありましたね。モケモケダンスとかやるやつ。私，アレ好きなんですけど。ゲームセンターといえば，最近はアタリのピットファイターというゲームがアツいです。残虐行為手当！（ああ，そうやって関係ないことでページが埋まってゆく……）

キャラクタのみでプログラムも基本的なコマンドしか使っていないようなので他機種にも簡単に移植できそうです。ただちょっとプログラムが気持ちスパゲッティ化してるような気がするので，移植するときにはどうにかしたほうがいいかも。

ああ，私の正体は地球を侵略するためにやってきたアルファベット星人だったのかあっ（おいおい）。

オマケは時計!?

ではでは，今月の2本目&3本目は東京都の阿妻さんの作品で「大時計.BAS」&「ジャンしゅー.BAS」です。

大時計&ジャンしゅー for X68000
(X-BASIC)
東京都　阿妻靖史

まず，大時計ですが早い話が時計です（ああああ……，説明になってない）。画面をいっぱいに使ったアナログ時計でちゃんと時間の修正もできるようになっています。以下のキーを押すことで，それぞれこんな機能になっています

A　修正モードに入る
←→　時間を合わせる（このとき，秒は0に戻る）
ESC　修正モードから抜ける。
リターン　時計の設定

なお，短針が赤いときが午前，緑のときが午後になっています。

そして「ジャンしゅー.BAS」。このプログラムは2つに分かれていまして，まず「JANKEN.BAS」を走らせてキャラクタを定義，そのあと本体の「JANSHU.BAS」を実行します。このゲームはジョイスティックを使うスクロールゲームです（残念ながらシューティングゲームではありません）。

自分がじゃんけんのグーチョキパーのい

リスト2　大時計.BAS

```
  10 /*     時計を表示するプログラム。名付けて時計を表示するプログラム。
  20 /*        名付けて  "時計を表示するプログラム "
  30 /*        そのまんまじゃねえかよ！
  40 int i,j,s0,s1,m0,m1,h,h0,h1
  50 /*色                     枠   文字 AM PM 長針  秒針
  60 /*int p_color(15)={0,16922,21124,5060,30852,12234,42280,0,
0,0,0,0,0,0,0,0}
  65 int p_color(15)={0,21160,42260,1984,63488,64168,61306,0,0,
0,0,0,0,0,0,52850}
  70 /*                  --------------------全角------------------
-  ----半角----
  80 str moji(12)={"","1","2","3","4","5","6","7","8","9","1
0","11","12"}
  90 screen 1,1,1,1:console ,,0
 100 for i=0 to 15
 120 palet(i,p_color(i))
 140 next
 150 apage(3)
 160 for i=7 to 14 :palet(i,hsv(int(rnd()*192+1),31,int(rnd()*1
0+22))):next
 170 for i=0 to 2000
 180   pset(int(rnd()*512),int(rnd()*512),int(rnd()*9+7))
 190 next
 200 apage(2)
 210 circle(255,255,250,1,,,384):circle(255,255,240,1,,,384):pa
int(255,10,1)
 230 for i=1 to 12
 240   symbol(232+135*cos(pi(i/6#-1/2#)),230+200*sin(pi(i/6#-1/
2#)),moji(i),2,2,2,2,0)
 250 next
 260 symbol(167,290,"X68000",2,2,2,2,0)
 270 repeat
 280   display(time$)
 290   if inkey$(0)="a" then adjust(time$)
 300 until 0
 310 end
 320 func display(tm;str)
 330 h=atoi(tm):m0=atoi(right$(tm,5)):s0=atoi(right$(tm,2))
 340 if s0<>s1 then second()
 350 if m0<>m1 then handm()
 360 endfunc
 370 func handm()
 380 apage(1)
 390 h0=h+12*(h>=12)
 400 an0=90-30*h1-m1/2:an0=an0-360*(an0<=0)
 410 circle(255,255,120,0,-an0,-an0,384)
 420 an0=90-6*m1:an0=an0-360*(an0<=0)
 430 circle(255,255,200,0,-an0,-an0,384)
 440 an0=90-30*h0-m0/2:an0=an0-360*(an0<=0)
 450 circle(255,255,120,3-(h>=12),-an0,-an0,384)
 460 an0=90-6*m0:an0=an0-360*(an0<=0)
 470 circle(255,255,200,5,-an0,-an0,384)
 480 m1=m0:h1=h0
 490 endfunc
 500 func second()
 510 apage(0)
 520 an0=90-6*s1:an0=an0-360*(an0<=0)
 530 circle(255,255,210,0,-an0,-an0,384)
 540 an0=90-6*s0:an0=an0-360*(an0<=0)
 550 circle(255,255,210,6,-an0,-an0,384)
 560 s1=s0
 570 endfunc
 580 func adjust(time;str)
 590 int a
 600 s0=0:second()
 610 repeat
 620   a=asc(inkey$(0))
 630   switch a
 640     case 52:m0=m0-1:if m0<0 then m0=59:h=h-1-24*(h=0)
 650             handm():break
 660     case 54:m0=m0+1:if m0>59 then m0=0:h=h+1+24*(h=23)
 670             handm():break
 680     case 13:time$=itoa(h)+":"+itoa(m0)+":"+"00"
 690     case 27:return()
 700   endswitch
 710 until 0
 720 endfunc
```

ずれかのキャラクタになっています。じゃんけんで勝てるものを取ると得点，あいこだと一定時間しびれてしまい，負けだとその場でゲームオーバーになってしまいます。そうそう，自分のキャラは次々と変わっていってしまうので間違えないように気をつけてくださいね。

プログラム自体は簡単だけどじゃんけんに勝つと自分が変わっていくというのはなかなかアイデア賞でしょう。じゃんけんという題材自体もわかりやすくてよいし。

ああ，それにしてもX68000もひさびさのピコピコゲームなのであります。今月はピコピコゲームの大バーゲンなのでありますよ，本当に。

ええ，実をいうとこのプログラムは投稿プログラムのおまけとしてついてきたもの(このパターン，この前もあったような)だったりするんですね。投稿のほうは迷路作成プログラムでして，おまけのほうがBASICだしプログラムも読んでもらえそうだなあ，ということでこっちを掲載することになってしまったのです。うーん，できれば迷路のほうのプログラムもいずれ載せたいんですけど。はてさてどうなることやら。

今月のお便り！

さてさて，また今月も投稿原稿からであります。大時計の阿妻さんの投稿原稿から。

「個人的な話になりますが，私は初めてOh!Xにハガキが載りました。また，電脳倶楽部も取り始めて間もないころに出したものが載っていました。で，思ったのですが雑誌の編集部の人は新しい読者には優しいのでしょうか，それとも新しい読者はまだその雑誌の影響を受けていないので編集部の人にとって新鮮に感じるのでしょうか？私は後者だと信じたいのですが……」

んーと，わかりませーん。だってハガキを選んでるのは私じゃないし，……ということです。それじゃあ，あまりに不親切ですか？　うーん，なんかこういう話って半年に1回ぐらいやってるような気もするんだけど，まいっか。それでは，手前みそながらショートプロの場合なぞ。

ショートプロの場合は古い読者の方でも常連さんでも載るときは載りますし，初めての人でも載らないときは載りません。

今回の阿妻さんがなんで載ったかというと，阿妻さんの気迫に負けたって感じです。阿妻さんは，メインになる迷路作成プログラム（これだけで8本も違うプログラムが入ってた)，迷路の解法プログラム，今回の大時計&ジャンしゅーとたくさんのプログラムを送ってきてくれたんです。投稿原稿も1つひとつ詳しく書かれていて，気合いが入ってましたしね。実をいうと本当は迷路も載せたかったんですが，残念ながらページの都合でこちらしか載せられなかったんです。できればそのうち迷路作成プログラムのほうも載せたいと思うんですけど，どうしようかな？

考えてみると最近の傾向として，プログラムを載せる基準は，「投稿に気合いが入っているかどうか」というところが重くなっているような気がします，私の場合。おかげさまでボチボチとショートプロ投稿も増えてきまして，掲載するプログラムを多数の中から選ばさせていただいてるわけですが，やっぱり“絶対に”載りたいと思っている人に載ってほしいなあ，などと思ってしまったりするわけです。

投稿してきてくださる方はみんな載りたいと思って出していますよね。しかし，原稿とかプログラムとかを見ているとおのずと気迫ってのは伝わってくるんです。阿妻さんのようにプログラムもドキュメントもいっぱい送ってくれたりとか，あっと驚くようなアイデアのプログラムとか，膨大な製作日誌がついてたりとか，私信がいっぱい書いてあったりとか，そして，プログラムがすごくきれいでエレガントに書かれてるプログラムとか。

まあ，でも，あんまりそんなこと気にしないでプログラムを作ってしまうのがベストなんじゃないでしょうかね。私の言うことなんか気にしないで。

リスト3 JANKEN.BAS

```
 10 screen 1,3,1,1:sp_init():sprite_pattern():end
 20 func sprite_pattern()
 30   dim char c(255)
 40   sp_def(0,c)
 50   c={
 60     &H0,&H0,&H2,&H2,&H2,&H0,&H2,&H2,&H2,&H0,&H0,&H0,&H0,&H0,&H0,&H0,
 70     &H0,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H2,&H2,&H2,&H2,&H2,&H0,&H0,&H0,
 80     &H0,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H2,&H0,&H0,&H0,
 90     &H0,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H2,&H0,
100     &H0,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H3,&H2,
110     &H2,&H2,&H2,&H2,&H2,&H2,&H2,&H2,&H3,&H2,&H3,&H3,&H2,&H2,&H3,&H2,
120     &H2,&H2,&H2,&H3,&H3,&H3,&H3,&H2,&H3,&H2,&H3,&H3,&H2,&H3,&H3,&H2,
130     &H2,&H3,&H3,&H3,&H3,&H3,&H3,&H2,&H3,&H2,&H3,&H3,&H2,&H3,&H3,&H2,
140     &H2,&H3,&H3,&H3,&H3,&H2,&H2,&H3,&H3,&H2,&H3,&H3,&H2,&H3,&H3,&H2,
150     &H2,&H3,&H3,&H3,&H2,&H3,&H2,&H3,&H3,&H2,&H3,&H3,&H2,&H3,&H3,&H2,
160     &H2,&H3,&H3,&H2,&H3,&H3,&H3,&H2,&H2,&H2,&H2,&H3,&H2,&H3,&H2,&H0,
170     &H0,&H2,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H2,&H3,&H2,&H2,&H0,
180     &H0,&H2,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H2,&H0,
190     &H0,&H0,&H2,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H3,&H2,&H0,&H0,
200     &H0,&H0,&H0,&H2,&H2,&H3,&H3,&H3,&H3,&H3,&H3,&H2,&H2,&H2,&H0,&H0,
210     &H0,&H0,&H0,&H0,&H2,&H2,&H2,&H2,&H2,&H2,&H2,&H2,&H0,&H0,&H0,&H0
220   }
230   sp_def(1,c)
240   c={
250     &H0,&H0,&HC,&HC,&HC,&H0,&H0,&H0,&H0,&H0,&H0,&HC,&HC,&HC,&H0,&H0,
260     &H0,&HC,&HD,&HD,&HD,&HC,&H0,&H0,&H0,&H0,&HC,&HD,&HD,&HD,&HC,&H0,
270     &H0,&HC,&HD,&HD,&HD,&HC,&H0,&H0,&H0,&HC,&HD,&HD,&HD,&HD,&HC,&H0,
280     &H0,&H0,&HC,&HD,&HD,&HD,&HC,&H0,&H0,&HC,&HD,&HD,&HD,&HC,&H0,&H0,
290     &H0,&H0,&H0,&HC,&HD,&HD,&HC,&H0,&H0,&HC,&HD,&HD,&HC,&H0,&H0,&H0,
300     &H0,&H0,&H0,&HC,&HD,&HD,&HD,&HC,&HC,&HD,&HD,&HD,&HC,&H0,&H0,&H0,
310     &H0,&H0,&H0,&HC,&HC,&HD,&HD,&HD,&HD,&HD,&HD,&HC,&HD,&HC,&H0,&H0,
320     &H0,&H0,&HC,&HD,&HD,&HC,&HD,&HD,&HD,&HD,&HC,&HD,&HD,&HD,&HC,&H0,
330     &H0,&H0,&HC,&HD,&HC,&HC,&HC,&HC,&HC,&HC,&HC,&HD,&HD,&HC,&HC,&H0,
340     &H0,&HC,&HD,&HD,&HD,&HD,&HC,&HD,&HD,&HD,&HC,&HD,&HC,&HC,&HD,&HC,
350     &H0,&HC,&HD,&HD,&HD,&HD,&HD,&HC,&HC,&HC,&HD,&HC,&HC,&HD,&HD,&HC,
360     &H0,&HC,&HD,&HD,&HD,&HD,&HD,&HD,&HC,&HD,&HC,&HC,&HD,&HD,&HC,&H0,
370     &H0,&H0,&HC,&HD,&HD,&HD,&HC,&HC,&HC,&HC,&HC,&HD,&HD,&HC,&H0,&H0,
380     &H0,&H0,&H0,&HC,&HD,&HC,&HD,&HD,&HC,&HC,&HD,&HD,&HD,&HC,&H0,&H0,
390     &H0,&H0,&H0,&H0,&HC,&HD,&HD,&HD,&HD,&HC,&HD,&HD,&HC,&H0,&H0,&H0,
```

いろいろ頭の中でぐだぐだ考えてても，考えてるだけじゃなにも始まんないんだもん。結局はプログラム作って送らなきゃ載らないんだし。作った者勝ちだよ，ね。作っているうちに気迫もおのずとわいてくるものだろうし。

……にしてもどうしましょう，迷路作成プログラム。できれば来月号あたりに載せたいんですけどね。うーん，うーん。ま，結果は来月号を見てのお楽しみっていうことで。

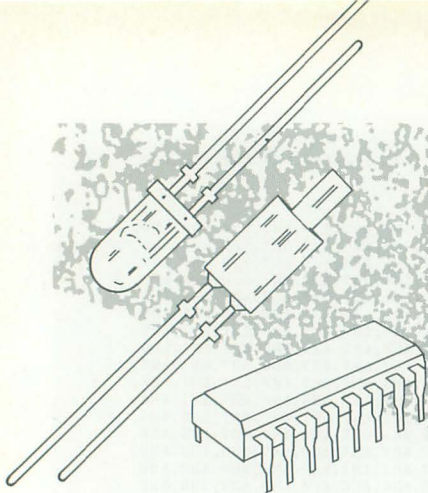
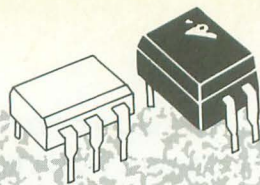
ではでは，そんなこんなでまた来月！

```
400       &H0,&H0,&H0,&H0,&H0,&HC,&HC,&HC,&HC,&HC,&HC,&HC,&H0,&H0,&H0,&H0
410     }
420     sp_def(2,c)
430     c={
440       &H0,&H0,&H0,&H6,&H6,&H0,&H0,&H6,&H6,&H0,&H0,&H6,&H6,&H0,&H0,&H0,
450       &H0,&H0,&H6,&H7,&H7,&H6,&H6,&H7,&H7,&H6,&H6,&H7,&H7,&H6,&H0,&H0,
460       &H0,&H0,&H6,&H7,&H7,&H6,&H6,&H7,&H7,&H6,&H6,&H7,&H7,&H6,&H0,&H0,
470       &H0,&H0,&H6,&H7,&H7,&H6,&H6,&H7,&H7,&H6,&H7,&H7,&H7,&H6,&H6,&H6,
480       &H0,&H0,&H6,&H7,&H7,&H7,&H6,&H7,&H7,&H6,&H7,&H7,&H7,&H6,&H7,&H6,
490       &H0,&H0,&H6,&H7,&H7,&H7,&H6,&H7,&H7,&H6,&H7,&H7,&H6,&H6,&H7,&H6,
500       &H6,&H0,&H6,&H6,&H7,&H7,&H6,&H7,&H7,&H6,&H7,&H7,&H6,&H7,&H7,&H6,
510       &H6,&H6,&H6,&H6,&H7,&H7,&H7,&H7,&H7,&H6,&H7,&H7,&H6,&H7,&H7,&H6,
520       &H6,&H7,&H7,&H6,&H6,&H6,&H6,&H7,&H7,&H7,&H7,&H7,&H6,&H7,&H7,&H6,
530       &H6,&H7,&H7,&H7,&H7,&H7,&H6,&H6,&H6,&H6,&H6,&H7,&H7,&H7,&H7,&H6,
540       &H6,&H7,&H7,&H7,&H7,&H6,&H6,&H6,&H7,&H7,&H7,&H6,&H6,&H7,&H7,&H6,
550       &H6,&H6,&H7,&H7,&H7,&H7,&H7,&H6,&H7,&H7,&H7,&H7,&H7,&H7,&H6,&H0,
560       &H0,&H6,&H6,&H7,&H7,&H6,&H6,&H6,&H6,&H6,&H6,&H6,&H7,&H7,&H6,&H0,
570       &H0,&H0,&H6,&H6,&H7,&H7,&H7,&H6,&H7,&H7,&H7,&H7,&H7,&H6,&H0,&H0,
580       &H0,&H0,&H0,&H6,&H6,&H7,&H7,&H6,&H7,&H7,&H7,&H7,&H6,&H0,&H0,&H0,
590       &H0,&H0,&H0,&H0,&H6,&H6,&H6,&H6,&H6,&H6,&H6,&H6,&H0,&H0,&H0,&H0
600     }
610     sp_def(3,c)
620 endfunc
```

リスト4 JANSHU.BAS

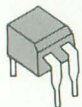
```
10 /*    じゃんけんシリーズ  NO. 1
20 /*    シューティングだぁ！
30 /*
40 /*
50 /*
60 /*
70 /*
80 /*
90 /*
100 /*
110 screen 1,3,1,1
120 int chi,phi=1,score,hsc
130 bg_set(0,0,1)
140 sp_disp(1)
150 console ,,0
160 repeat
170   title()
180   for phi=1 to 10
190     ge=game()
200     if ge=0 then gover():break
210     print phi;"面をクリアしました"
220     for chi=0 to 30000 :next:cls
230   next
240 until phi>=10
250 /*
260 ending()
270 end
280 /*
290 /*
300 func int game()
310 char shini,at,joy
320 int movex=256,movey=496,scrl,ptn1,mptn,stopf,ppp
330 bg_fill(0,0)
340 for i=0 to 3
350   for ppp=0 to 63
360     scrl=ppp*16:score=score+1
370     if (scrl mod 16)=0 then bputs(scrl/16)
380     if stopf=0 then {
390       joy=stick(1)
400       switch joy
410         case 1:if movey<496 then movey=movey+16
420                if movex>128 then movex=movex-16
430                break
440         case 2:if movey<496 then movey=movey+16
450                break
460         case 3:if movey<496 then movey=movey+16
470                if movex<368 then movex=movex+16
480                break
490         case 4:if movex>128 then movex=movex-16
500                break
510         case 6:if movex<368 then movex=movex+16
520                break
530         case 7:if movey>48  then movey=movey-16
540                if movex>128 then movex=movex-16
550         case 8:if movey>48  then movey=movey-16
560                break
570         case 9:if movey>48  then movey=movey-16
580                if movex<368 then movex=movex+16
590                break
600       endswitch
610     }
620     at=atari(movex,movey,scrl,mptn)
630     switch at
640       case 0:break
650       case 1:score=score+10:break
660       case 2:stopf=7:break
670       case 3:return(0)
680     endswitch
690     sp_move(0,movex,movey,mptn)
700     bg_scroll(0,0,1023-scrl)
710     if stopf>0 then stopf=stopf-1
720     if (scrl mod 50)=0 then {
730       if mptn=3 then {
740         mptn=1
750       } else {
760         mptn=mptn+1
770       }
780     }
790   locate 0,0 :print "SCORE";score
800   next
810 next
820 return(1)
830 endfunc
840 /*
850 func bputs(s;int)
860 int x,ran
870 if s>62 then s=s-64
880 for x=0 to 15
890   ran=(rand() mod (31-phi*2))
900   if ran>3 then ran=0
910   bg_put(0,x+8,62-s,256+ran)
920 next
930 endfunc
940 /*
950 func pause()
960 str ten[1]
970 repeat
980   ten=inkey$
990 until ten=" "
1000 while ten<>" "
1010   ten=inkey$
1020 endwhile
1030 endfunc
1040 /*
1050 func int atari(mx;int,my;int,sc;int,mp;char)
1060 char sts
1070 int i,j,ret,bgx,bgy
1080 bgx=(mx+8)/16:bgy=((my-sc+1024) mod 1024)/16
1090 sts=bg_get(0,bgx,bgy) mod 256
1100 if sts=0 then ret=0 else ret=(mp-sts+3) mod 3+2
1110 if ret=4 then ret=1
1120 switch ret
1130   case 3:for i=0 to 5
1140            sp_on() :for j=0 to 1000 :next
1150            sp_off():for j=0 to 1000 :next
1160          next:break
1170   case 1:bg_put(0,bgx,bgy,256)
1180 endswitch
1190 return(ret)
1200 endfunc
1210 /*
1220 func gover()
1230 cls:bg_set(0,,0):sp_off()
1240 locate 22,14:print "GAME OVER"
1250 locate 22,16:print "SCORE ";score
1260 if hsc<score then {
1270   hsc=score
1280   locate 21,18:print"PLAYER HIGH SCORE!"
1290 }
1300 for ml=0 to 30000 :next
1310 endfunc
1320 /*
1330 func ending()
1340 print "Congraturations!!!"
1350 endfunc
1360 func title()
1370 cls
1380 score=0
1390 bg_set(0,,0)
1400 sp_off()
1410 symbol(80,64,"JANSHOO",4,8,2,64478,0)
1420 locate 20,20:print"PUSH TRIGGER"
1430 locate 20,22:print"HIGH SCORE   ";hsc
1440 repeat
1450 seed=seed+1
1460 if seed=32767 then seed=1
1470 srand(seed)
1480 until strig(1)<>0
1490 cls:wipe():sp_on():bg_set(0,0,1)
1500 endfunc
```

ハードウェア工作入門《9》

センサー回路その3

Misawa Kazuhiko
三沢 和彦

2回にわたってセンサーの基礎について見てきましたが，概要は理解していただけたでしょうか。今回はアルコールセンサーと光センサーの2つを実際に工作してみます。これまでの記事も参照しながら進めていきましょう。

今月はお待ちかね，センサー回路の製作実習です。先々月，先月と理論編が2回続きで待ちくたびれてしまったかもしれませんが基礎をばっちり固めた上で実際に工作に移るほうがよく理解できると思います。

今回の回路はアルコールセンサーと光センサーの2種類ですが，光センサーのほうはオペアンプの増幅回路が2段ついているだけに外付け部品の点数がやや多くなっています。まずはアルコールセンサーのほうから簡単に片づけてしまいましょう。

アルコールセンサーの製作

部品表は表1のとおり，本体のアルコールセンサーとA/Dコンバータに接続するためのステレオミニプラグ，そして1kΩの抵抗1本だけです。今回のアルコールセンサーは日本セラミック株式会社製のNGSX-03というタイプのもので，半導体素子を使用しアルコールに対して特に高感度を示すものです。このアルコールセンサーはいつものT-ZONEパーツセンターで入手したのではなく，秋月電気通商（☎03-3700-5212）という店で購入しました。秋月電気通商はハードウェア工作マニアのなかでは馴染みのある店で，特に特殊な掘り出し部品が格安で手に入る店です。以前，本誌でもストロングタイプのハードマニアである茱野氏が，ガイガーカウンタキットを購入したのもこの秋月でした。

さて，外付け部品は抵抗線1本だけですから，基板に組み立てるほどのものでもありません。そこで，外付けの抵抗線はステレオミニプラグの中に取り付けることにします。ステレオミニプラグの端子の様子は本誌1990年11月号の図5にあります。問題の1kΩ抵抗は先々月の回路図（1月号の図5）によるとA/Dコンバータの入力端子とGNDとの間に入っています。ステレオミニプラグでいうと真ん中の軸上に出ているのが電圧入力，一番外側の金属部がGNDに対応しているので，図1のように橋渡しのような形で取り付けます。これさえできれば，あとはビニール線でアルコールセンサーの各端子とハンダ付けすれば完成です。

アルコールセンサーの端子を図2に示します。センサーの電極部分の両端（端子1と4），ヒーターの両端（端子2と5）の計4カ所をステレオミニプラグに接続しますが，実体配線図を参考にして間違えないようにしてください。特に5Vの線は1本で電極とヒーターの両方にハンダ付けするので注意が必要です。

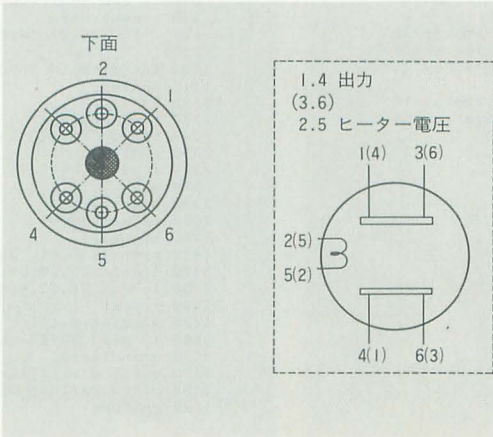
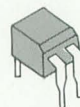



図1 アルコールセンサー外付け抵抗の取り付け

図2 アルコールセンサーの端子

光センサーの製作

光センサーのほうは部品数も多いので，基板上に組みます。今回使用した基板はいつものサンハヤトICB-87です。実体配線図（図3）をよく見て間違えないようにしてください。ここで，まず**バグのお詫び**です。1月号の回路図（図6）で，オペアンプLM358の＋5Vが4番ピン，GNDが8番ピンになっていますが，これは逆で，正しくは**＋5Vが8番ピン，GNDが4番ピン**です。今月の実体配線図では正しいほうになっていますので，安心して配線してください。

この入門記事の回路は，すべて実際に組み立てて動作チェックしていますので，完成品が全く動かない心配はありません。ただし，回路図か実体配線図のどちらかを図面に描く際にミスすることがあるかもしれないので，もし回路図と実体配線図とを見比べておかしいと思われるときは連絡してください（実際，基本I/O回路のときには読者の方からミスを指摘していただきました）。

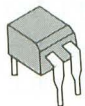
では，作業に移りましょう。手順としてはいつものようにまずICソケットを取り付けます。3番ピンと4番ピンはGNDに直結，8番ピンは＋5Vに直結なので，足を折り曲げてそれぞれのラインにもハンダ付けします。先ほど述べたとおり，LM358のGNDは4番ピン，＋5Vは8番ピンなので，注意してください。

次にフォトダイオードをハンダ付けしま

す。フォトダイオードにはアノードとカソードの2つの極性があって，カソードをオペアンプの＋入力（2番ピン）に，アノードをGNDにつなぎます。フォトダイオードから出ている2本のリード線のどちらがアノードでどちらがカソードかはフォトダイオードの型番によって違うので，部品を購入するときにその店でチェックしておいてください。今回私が使ったフォトダイオードは光電子というメーカーのSP1-CL3というタイプのもので，これはT-ZONEで入手しました。もしこれと同じタイプのものを使うのであれば，その端子は図4のようになっています。もしここで逆につないでしまうと，A/Dコンバータの入力が負の電圧になってしまい，読み取り値が全部0になってしまいます。

あとはほとんど問題のない部品です。1段目の電流増幅器の1MΩの抵抗と200pFのコンデンサを並列にしてオペアンプの1番と2番の間につなぎます。2段目の非反転増幅器では，まず10kΩの抵抗を6番とGNDの間につなぎます。この10kΩをGNDに落とすところでは，基板の表側に抵抗の足を回して立体交差させていますので，図5を参考にして配線してください。

また，可変抵抗は基板取り付け型のトリマー抵抗器というものを使っています。これはマイナスドライバーでねじを回して抵抗値を調整するものです。テレビなどのボリュームツマミや前回の簡易ジョイスティックと違って，頻繁に調整する必要がない場合にはこのトリマー抵抗をよく使います。写真の完成品では少し高級な多回転型のトリマー抵抗を使っているので，少し大きめですが，実体配線図ではごく普通のトリマーにしておきました。ジャンパ線は2カ所忘れずに配線してください。

最後にいつものステレオミニプラグに＋5V，A/Dコンバータの電圧入力，GNDの3本を結線して完成です。

動作チェック

今回のセンサー回路はA/Dコンバータに接続することを前提にしていますので，A/Dコンバータを使った動作チェックが最も簡単です。それには，1990年11月号のサンプルプログラムが必要なので，ぜひ入力してください。といっても，すでにA/Dコンバータを製作した人はもう入力ずみでしょう。また，12月号のadread関数を組み込んだ人は，今月のリスト1だけでもOKです。

ではまず，アルコールセンサーから動作チェックです。今回製作したアルコールセンサーのステレオミニプラグをA/Dコンバータボードに差し込んでください。ここでもう一度注意しておきますが，**A/Dコンバータボード本体をジョイスティックポートに差し込んだあとで，このプラグを抜き差しするとADC0832が暴走します。**このときはまだA/Dコンバータボードを X68000 につながないでおいてください。

次にBASICを起動し，サンプルプログラムをRUNさせてください。すると画面の左側に255が並んでいるでしょう。この状態でA/Dコンバータボードを本体につないでください。表示が0まで下がってからだんだん増えていき，40ぐらいまで上がったら今度はゆっくり下がっていきます。そして

表1

アルコールセンサー		
アルコールセンサー(NGSX-03)	1個	1,700円
1kΩ抵抗	1本	10円
ステレオミニプラグ	1個	100円
ビニール配線材	少々	
光センサー		
IC用基板(サンハヤトICB-87)	1枚	90円
フォトダイオード(SP1-CL3)	1個	150円
オペアンプLM358	1個	60円
8ピンICソケット	1個	30円
1MΩ抵抗	1本	10円
10kΩ抵抗	1本	10円
100kΩ半固定トリマー抵抗	1個	50円
200pFスチロールコンデンサ	1個	
ステレオミニプラグ	1個	100円
ビニール配線材	少々	

図3 実体配線図

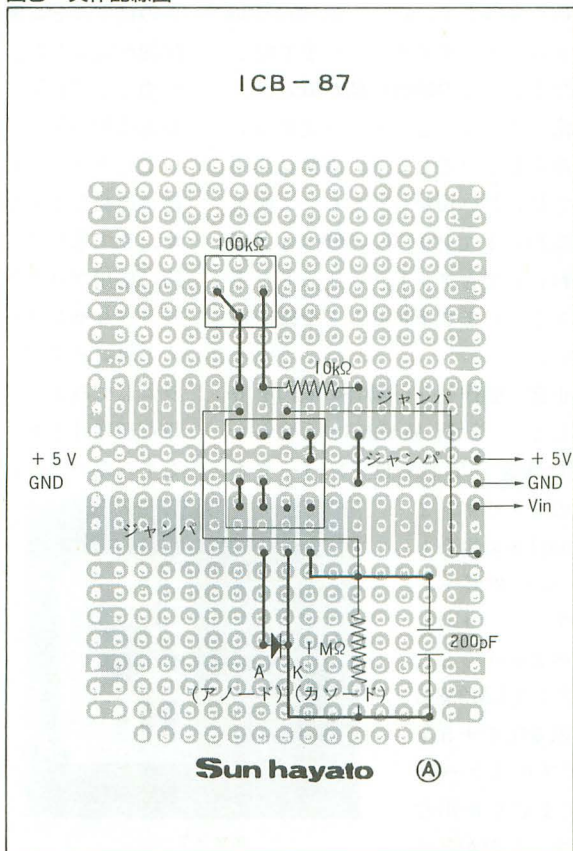



図4 フォトダイオードSP1-CL3

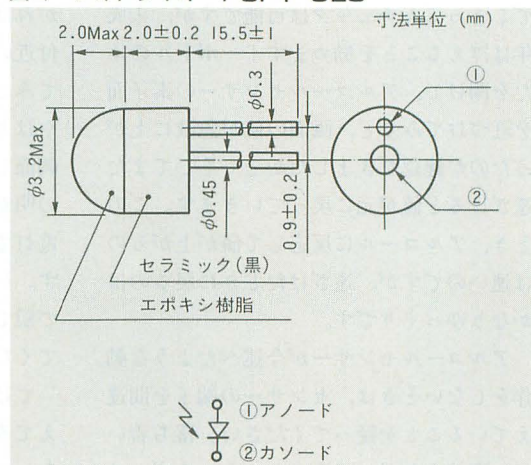
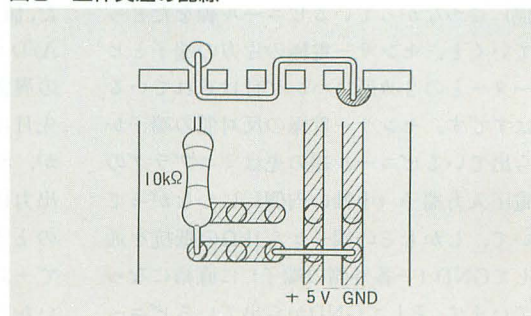
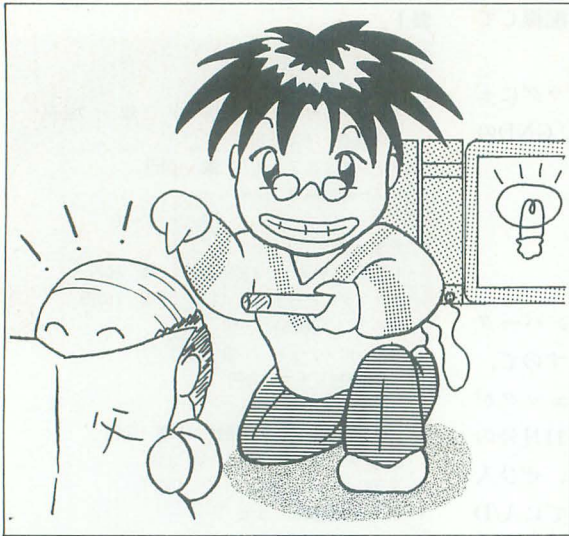



図5 立体交差の配線

最終的には30ぐらいで落ち着きます。いま挙げた値はおおよその目安で，実際に使っているセンサーによってバラつきがありますが，基本的には大きい値からゆっくり一定値に近づいていくような動きをすれば正常に動作しているといってよいでしょう。全体の動きは画面上でグラフとしてプロットされるので容易に確認できます。

ではいよいよアルコールに対する反応のチェックです。ウイスキーか日本酒のボトルを準備してください。これ幸いにと飲んでしまってもチェックは可能ですが，未成年は控えることを勧めます！　ボトルのふたを開けて，アルコールセンサーの素子面を近づけてみると，画面の値が急激に上がったのが確認できましたか？　そしてまた遠ざけると値が元に戻っていきます。このとき，アルコールに反応して値が上がるのは速いのですが，遠ざけたときに戻るのはかなりゆっくりです。

アルコールセンサーが今述べたような動作をしないときは，センサーの端子を間違えていることを疑ってください。落ち着いて，配線をたどってみましょう。まず，ステレオミニプラグの＋５Ｖ端子（中軸の外側）につながっているビニール線をたどっていくと，センサー電極の片方の端子とヒーターとの２カ所にハンダ付けされているはずです。センサー電極の反対側の端子から出ているビニール線の先はミニプラグの電圧入力端子（中軸の内側）につながっていて，しかもその端子から1kΩの抵抗を通してGND（一番外側の端子）に直結になっています。そしてGNDから出ているビニール線の先はヒーターのもう片方の端子につながっているはずです。

今述べた順番でチェックしてどうしても配線に間違いがないのに，正常に動作しない時にはA/Dコンバータボード自体のチェックをやり直してください。

では，次に光センサー回路のほうの動作チェックです。光センサー回路を先にA/Dコンバータボードに接続しておいて，サンプルプログラムをRUNさせてから，本体につなぐのは先ほどと同じ手順です。今度は，つないだ直後からある一定値を示すと思います。フォトダイオードにSP1-CL3を使っている限りでは，この値は３ぐらいから190ぐらいの間にあるはずです。

まずこの値が10以下のときは懐中電灯を持ってきて，フォトダイオードを近くで照らしてみてください。照らした瞬間，値が増えれば正常です。懐中電灯で値が上がるのでしたら，試しに倍率調整の半固定抵抗を回してみてください。値が100近くまで上がればしめたものです。また一方，値が190付近のときはフォトダイオードを手で隠してみてください。この瞬間に値が減れば，やはり正常です。このときも半固定抵抗で調節してみるとよいでしょう。調整は部屋の明かりでだいたい10から20ぐらい，懐中電灯で直接照らして190ぐらいがベストです。いずれにしてもフォトダイオードを手で隠して３ぐらいまで下がることを確かめてください。

では，正常に動作しない場合の原因を考えてみましょう。まず，値が255から変わらないときは間違いなくA/Dコンバータ本体のほうのミスです。また，値が200以上の場合もA/Dコンバータボードの異常です。というのも，先月は述べませんでしたが，オペアンプLM358の出力は電源電圧が＋５Ｖのときはフルスケールで＋3.8Ｖまでしか出ないからです。したがって，光センサー回路のほうでフルスケールまで出力が出ているときは，

3.8/5×256＝194.56

がA/D変換後の最高値になるはずなのです(先ほどまでのおよそ190というのはこのことである)。あとは，ごくまれにLM358の出力７番ピンが＋５Ｖとショートしている配線ミスがあります。

次に，値が０付近で，いくら光を当てても変化しない場合はまずフォトダイオードを逆に接続していることが考えられます。フォトダイオード自体の不良ということも考えられなくもないのですが，まずはフォトダイオードの配線ミスを疑ってください。あとは，LM358の出力７番ピンがGNDとショートしている配線ミスがあります。

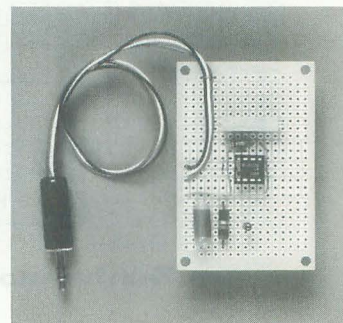
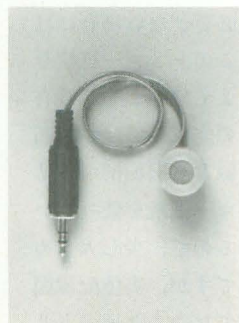
そして，値が190付近で，いくらフォトダイオードを手で隠しても値が下がらないときは，オペアンプ周りの配線ミスが考えられます。たとえば，オペアンプの－入力と＋入力とを逆にしてしまうと出力が飽和します。同様に，負帰還抵抗（1MΩおよび100kΩVR）の配線を間違えても出力は飽和します。

＊

いかがでしたか？　異常な動作をするときのほとんどは配線ミスが原因です。一度配線が完了したら，ひとまず休憩して気分転換をしてから，配線チェックをしたほうが効率的です。どうしても正常に動作しないで行き詰まってしまったら，封書で編集部に連絡してください。誌上でお答えします。さて来月は，今月製作したセンサー回路の応用プログラムを載せる予定です。アルコールセンサーのほうは皆さんも想像つくと思いますが，飲酒チェッカーをプログラムしてみます。光センサーのほうは，簡単な照度計とあとひとつなにかプログラムしてみたいと思いますので，お楽しみに。

光センサー

アルコールセンサー

THE SENTINEL

●童心に帰るアクションゲーム

寒い日々が続きますがいかがお過ごしですか。雪国では子供たちが元気に雪合戦をしていることでしょう。今月は，そんな季節感あふれるゲームをお届けしましょう。

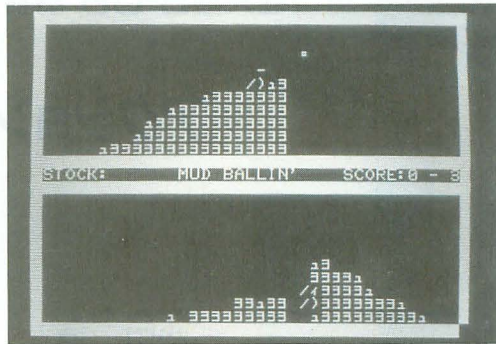
届いたゲームは“MUD BALLIN'”。地面に転がっている雪玉(本当は泥玉だけど……)を拾っては投げ，拾っては投げ，一定の数だけ相手に当てれば勝ちという単純なゲームです。このゲームの難しさは，投げた玉が放物線を描いて飛んでいくという点にあります。しかも，どの程度強く投げるのかはXキーを押している時間で決まり，なかなか自分の思いどおりの場所に投げることはできません。

このゲームは文字だけを使って，しかも40字モードで画面が構成されています。40字の文字画面で放物線？　と疑問に思われるのもごもっとも。通常なら描き切れなくなりそうなところですが，画面を縦に2分割するというアイデアで乗り切りました。画面上半分には自分が，画面下半分には相手が表示され，投げた玉はいったん画面の外に消えてから再び戻ってくるのです。これが距離感をいっそうつかみづらく，ゲームを面白くしています。

第104部
MUD BALLIN'

●S-OSの#LOCに注意

制作者の柴田さんが最初に本誌に登場したのは1988年2月号でした。X1とBASICで行ったCGアニメーションフィルムの作成記事です。迷路の中を駆けていく人のアニメーションは非常にインパクトがあり，これが本当にX1とBASICで作られたものなのかと我が目を疑ったほどです。また，1989年12月号で登場したX1/turbo用のアクションゲームACTIVE UNITの作者でもあります。いずれも既成の概念への挑戦といった心意気を感じることができる作品ですが，その心髄は今回の作品にも十分に生きているといえるでしょう。

投稿されてきたプログラムはX1で作成されたものだったのですが，編集部で手を加え完全S-OS対応としました。その際に気づいたのですが，柴田さんに限らずかなりS-OSを使い込んでいらっしゃる方の投稿でも，カーソル位置を指定する#LOCルーチンが誤って使われていることが少なくありません。仕様にあるとおり，#LOCルーチンはAFレジスタを保存しません。機種によっては呼び出し前と同じものが返ってきますが，破壊するS-OSも存在します。投稿の際には注意してください。

●S-OSの系譜(19)

S-OSのモニタにシェルライクな機能を付加する改造が行われた1987年5月号には，MZ-700用“SWORD”をQD対応にする記事も掲載されました。MZ-1500で初めて採用されたQDは，カセットテープの技術を生かした安価なフロッピーディスクといった感じのもので，片面64Kバイトと小容量ながらカセットテープとは比較にならないデータ入出力スピードを持っていました。これをMZ-700につなぎ，S-OSでサポートしようというのです。これによってプログラムの起動が高速になるだけでなく，プログラムの開発時間が短縮されることになりました。

続く1987年6月号では待望のFuzzyBASICコンパイラが発表されました。これはスタッフの作品ではなく投稿，しかも制作者の石上氏は当時16歳の高校生でした。簡単なベンチマークの結果も良好で，インタプリタのFuzzyBASICで作成したプログラムを，アセンブラで作ったプログラムに近いスピードで実行させることが可能となったのです。S-OSに本格的な高級言語のコンパイラがサポートされたことの意義は大きく，同時にインタプリタでデバッグを済ませてからコンパイラで高速化するという，ユーザーフレンドリな開発環境が用意できたことにもなります。あらためてユーザーパワーのすごさを実感した投稿でした。

同時にこの6月号では瀧山氏によるZEDAのバージョンアップ版，ZEDA-3も発表されました。先に1986年の9月号でアセンブル速度の向上と，メモリに入り切らないプログラムをアセンブルする改良が加えられていましたが，今回のZEDA-3ではより高速に，よりメモリを有効に使う方向でバージョンアップがなされました。また，ZEDA-3発表までに報告されていたバグもこのとき取り除かれました。初代ZEDAからZEDAの改造，そしてZEDA-3，REDA，WZDと，S-OSは実に5本ものアセンブラを供給してきました。そしてそのアセンブラたちが，次代のアプリケーションを生み出し，S-OSの歴史を刻んできたのです。

全機種共通S-OS“SWORD”要

アクションゲーム MUD BALLIN'

これまでのS-OS用ゲームとはひと味違うアクションゲームです。Rythm to Traceを覚えているでしょうか? あの柴田君がS-OSに登場です。キャラグラの表現もなかなか。どこを取っても斬新なアイデア,オリジナリティが光ります。

Shibata Atsushi
柴田 淳

S-OSの標準キャラクタを組み合わせて作った人型のキャラクターを操る,雪合戦ならぬ泥合戦型アクションゲームです。キャラクタグラフィックの味のあるアニメーションと,滑らかな弾道を描いて飛んでくる玉など,これまでのS-OS用ゲームとはひと味違った内容になっています。

入力&操作方法

各機種のマシン語モニタまたはMACINTO-Cなどのマシン語入力ツールからリスト1を打ち込みます。チェックサムやCRCチェックバイトを確認のうえ,3000_H番地から4100_H番地までを実行開始番地3000_Hでセーブしてください。実行はS-OSのモニタから,

#J3000

です。

ゲームが始まるとタイトルデモ画面が表示されます。まずは画面を見てください。上下に2分割されていますが,上の画面の右端と下の画面の左端はつながっています。デモ画面ではその横長のフィールドを人間(に見えるかな?)がちょこまかと動きながらタイトル文字を崩していますので,しばらく見惚れて,ゲームを始める気になったらなにかキーを押してください。

最初に地形を作っておかなくてはなりません。REGULARモードではいくつかの地形がプリセットされているので初めはこちらを選ぶのがよいでしょう。このモードでは地形は9種類用意してあります。数字キーを押すとその番号の地形が画面に呼び出されますのでお好きなものを選んでください。ここで“Z”キーを押すと下の画面を選択できます。相手側の地形も同様に選択したらリターンキーで確定します。ここで確認をとってきますので,よければ“N”以外のキーを押してください。

次にコンピュータのレベルを設定します。1はまあまあ簡単,最初から5などを選ぶとあっというまに集中砲火を浴びて終わってしまいますので,最初は1を選択するのが無難でしょう。慣れてきたらだんだんに上げていってください

これでゲームが始まります。

ルールは明快。相手に先に8発の泥玉をぶつけたほうが勝ちです。4,6キーで左右にキャラクターを移動させ,Zキーで泥玉を取り,Xキーで相手にぶつけます。

キャラクターは1/2段分(泥玉1個分)の低い段差なら泥壁を昇っていけますが,それ以上だと移動できなくなります。高いところから勢い余って落ちてしまうと元に戻れなくなり,相手の泥玉の餌食になることがあります。

すでに手に泥玉を持っているときにZキーを押すと今度は泥玉を地面に置くことができます。これを利用すれば段差を埋めて泥壁を昇ったり,バリケードを作ったりなどという高等戦略を立てることもできるでしょう。

攻撃する際には闇雲に投げてもそう当たるものではありません。泥玉の飛距離を制御することが勝利の秘訣です。このゲームでは投げるときのXキーだけで飛距離まで調整できるのです。Xキーを長く押していると泥玉は遠くに飛びます。短く押すと少ししか飛びません。最初は難しいと思いますが,このタイミングをつかめば勝利は目前です。

そのほか,泥玉を持っているときに限り2キーでその場にしゃがむことができます。相手の攻撃が激しくなってきたらこれで急場をしのいでください。ただし,このとき持っている泥玉は失ってしまい,短い時間しかしゃがんでいられませんのでタイミングが大切になります。

ゲームの進行状態によっては双方とも玉を投げ尽くしたり,動けなくなったりと,どうしようもなくなることがあります。こういうときにはGキーを押してください。タイトル画面に戻ります。

また,ここで挙げたキー操作はテンキーを持つ機種にあわせてありますので扱いづらい方は表1の部分を好みのキャラクタコードで書き換えてください。

表1 キーデータアドレス

左移動	: $30A0_H$
右移動	: $30A5_H$
しゃがむ	: $309B_H$
取る	: $30AF_H$, $30C6_H$
投げる	: $30B4_H$, 3260_H, $326E_H$, 3298_H

コンストラクション

ひととおり操作を覚えたら,とにかくやってみるのが上達の早道でしょう。また,地形の具合が勝敗に大きくかかわってきますので,最初はコンピュータをプリセットの6番の地形にしておくといいかもしれません。遊んでいるうちにそれぞれの地形が持っている特徴や癖がわかるようになります。そうなったらしめたもの。今度はオリジナルの地形を作ってみましょう。

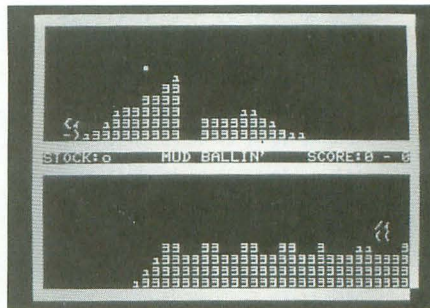
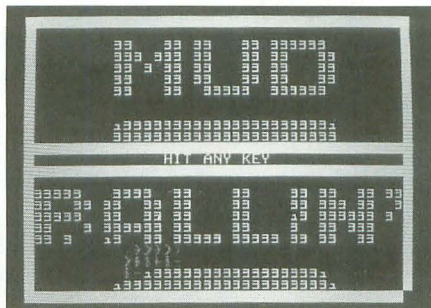
ゲーム開始時の地形選択のところでCONSTRUCTIVEモードを選ぶとコンストラクションモードになります。

操作は簡単。4と6のキーでカーソルを左右に動かし2,8のキーで地形を上下させていきます。一度REGULARモードで地形を呼び出してそれからCONSTRUCTIVEモードで部分的に修正することもできます。

こうして工夫を重ねた面を相手に，苦難を乗り越えて勝ったときの感慨はひとしおのものがあります。

最後に

このゲームを作るきっかけになったのは遊びで作ったキャラグラの「走る人」でした。なかなかよくできていたので，これをなにかに使えないかなと考えているうちに「雪合戦」とあわさってついにこういうかたちになりました。

玉の飛び方の滑らかさにもいくつか秘密があって，そこも我ながら気にいっている点です。玉の飛び方とか操作感覚というのは，ときにはルール以上に強くゲームの性格を表すことがあるものです。

早く泥玉を取りたくてあたふたすることとか，ふわふわ飛んでくる玉に当たって倒れる悔しさなどを味わっていただければ，作者としては本望です。

これまで発表されているS-OSやMZ-700用の一連のゲームを見ていて，キャラクタグラフィックを使ったゲームを作るのにずっと憧れていました。ただ，それにみあったアイデアはなかなか思い浮かばなかったのですが，今回やっとこのゲームを発表できることになって，かつての名作たちと少しでも並べて語られるようになればと思っています。

```
3000 CD 4B 3C ED 73 FF 40 ED : E0
3008 7B FF 40 CD 51 3C CD 8F : 70
3010 38 CD 5D 3C CD DB 35 CD : 48
3018 11 39 CD 51 3C CD 3D 30 : DE
3020 CD 65 33 CD 2E 3B CD AF : 17
3028 39 01 00 08 CD 9D 3C CD : B5
3030 AF 39 CD 9D 3C C3 1D 30 : 9E
3038 ED 7B FF 40 C9 21 07 0C : A4
3040 CD 1E 20 3A 2D 40 06 20 : D8
3048 B7 28 02 06 6F 78 CD F4 : 8F
3050 1F 3A 29 40 B7 1F 6F 3A : 41
3058 2A 40 67 24 24 CD 6A 3B : 8B
3060 FE 20 C2 81 30 2C CD 6A : F4
3068 3B FE 20 C2 81 30 2A 29 : 1F
3070 40 B7 CB 1D 3E 00 CD 3F : 29
3078 3B 24 7C 32 2A 40 C3 4A : 84
---------------------------------
SUM: B4 23 80 2F 5D DF DF D6 6301

3080 32 3A 2B 40 FE 32 D2 14 : ED
3088 33 FE 20 D2 F3 32 FE 07 : 4D
3090 D2 5E 32 CD D0 1F F5 08 : 1B
3098 F1 08 FE 32 CA 14 33 FE : 38
30A0 34 CC 76 31 FE 36 CC DA : 81
30A8 31 FE 47 CC 07 30 FE 5A : D1
30B0 CC C2 30 FE 58 CC 5E 32 : 70
30B8 08 32 37 40 CD C7 1F 38 : 9C
30C0 30 C9 3A 37 40 FE 5A C8 : CA
30C8 3A 2D 40 FE 01 CA 1F 31 : C0
30D0 3A 29 40 B7 1F 6F 3A 2A : 4C
30D8 40 67 3A 2B 40 FE 05 38 : 87
30E0 03 2D 18 02 2C 2C 24 CD : 93
30E8 6A 3B FE AD CA 01 31 FE : 4A
30F0 20 C8 25 CD 6A 3B FE D6 : 53
30F8 C8 FE 20 20 04 24 CD 6A : 65
---------------------------------
SUM: 9A 10 EE FF B9 51 17 25 8729

3100 3B FE D6 CA 0C 31 FE AD : C1
3108 CA 14 31 C9 3E AD CD 8C : 1C
3110 3B C3 19 31 3E 20 CD 8C : FF
3118 3B 3E 01 32 2D 40 C9 3E : 20
3120 00 32 2D 40 3A 29 40 B7 : F9
3128 1F 6F 3A 2A 40 67 3A 2B : FE
3130 40 FE 05 38 03 2D 18 02 : C5
3138 2C 2C 24 CD 6A 3B FE AD : 99
3140 CA 70 31 FE 20 CA 57 31 : DB
3148 25 CD 6A 3B FE 20 CA 65 : E4
3150 31 3E 01 32 2D 40 C9 24 : FC
3158 CD 6A 3B FE 20 CA 57 31 : E2
3160 FE AD 28 01 25 FE AD CA : 6E
3168 70 31 3E AD CD 8C 3B C9 : E9
3170 3E D6 CD 8C 3B C9 2A 29 : C4
3178 40 B7 CB 1D 3E 00 CD 3F : 29
---------------------------------
SUM: DF 2E 86 25 72 7D 11 7A BEB1

3180 3B 3A 29 40 3D FE 98 D2 : 83
3188 C4 31 FE 02 DA C4 31 CB : 8F
3190 47 20 06 32 29 40 C3 C4 : 8F
3198 31 6F B7 CB 1D 3A 2A 40 : E3
31A0 67 CD 6A 3B FE D6 CA C4 : 3B
31A8 31 24 CD 6A 3B FE D6 CA : 65
31B0 C4 31 FE AD 20 07 3A 2A : 2B
31B8 40 3D 32 2A 40 3A 29 40 : BC
31C0 3D 32 29 40 3A 2B 40 FE : 7B
31C8 05 30 02 3E 04 3C FE 07 : BA
31D0 20 02 3E 05 32 2B 40 C3 : C5
31D8 4A 32 2A 29 40 B7 CB 1D : AE
31E0 3E 00 CD 3F 3B 3A 29 40 : 28
31E8 3C FE 98 D2 4A 32 47 3A : A1
31F0 30 40 D6 05 B8 DA 4A 32 : 59
31F8 78 CB 47 28 06 32 29 40 : 53
---------------------------------
```

```
SUM: E1 F8 60 A5 E9 12 E5 6A 4D50

3200 C3 2F 32 6F B7 CB 1D 3A : 6C
3208 2A 40 67 2C CD 6A 3B FE : 6D
3210 D6 CA 4A 32 24 CD 6A 3B : B2
3218 FE D6 CA 4A 32 FE AD 20 : E5
3220 07 3A 2A 40 3D 32 2A 40 : 84
3228 3A 29 40 3C 32 29 40 3A : B4
3230 2B 40 FE 05 38 0A 3A 29 : 13
3238 40 CB 87 32 29 40 3E 00 : 6B
3240 3C FE 05 20 02 3E 01 32 : D2
3248 2B 40 21 3C 3D 3A 2B 40 : AA
3250 85 6F 2D 7E 2A 29 40 B7 : E9
3258 CB 1D CD 3F 3B C9 08 3E : 3E
3260 58 08 3A 2B 40 FE 0E D2 : E3
3268 89 32 3A 37 40 FE 58 C8 : 8A
3270 3A 2D 40 B7 C8 3D 32 2D : C2
3278 40 3A 2B 40 32 2C 40 3E : C1
---------------------------------
SUM: 7F E8 9B 3C C8 74 9D A2 D705

3280 0D 32 2B 40 3E 00 32 2E : 48
3288 40 2A 29 40 B7 CB 1D B7 : 29
3290 1F CD 3F 3B CD D0 1F FE : 20
3298 58 20 08 3A 2E 40 C6 06 : F4
32A0 32 2E 40 3A 2B 40 3C FE : 7F
32A8 12 CA B9 32 FE 14 20 03 : FC
32B0 3A 2C 40 32 2B 40 C3 F0 : F6
32B8 32 21 8F 40 01 07 00 7E : A8
32C0 B7 28 04 09 C3 BF 32 3E : DE
32C8 01 77 23 3A 29 40 B7 CB : C0
32D0 1F 3C 77 23 3A 2A 40 B7 : 50
32D8 17 77 23 3E 01 77 23 3E : C8
32E0 00 77 23 77 3A 2E 40 D6 : 8F
32E8 05 23 77 3E 12 32 2B 40 : 8C
32F0 C3 B8 30 2A 29 40 B7 CB : C0
32F8 1D 3A 2B 40 B7 1F C6 03 : 61
---------------------------------
SUM: 47 6C 19 96 98 D5 87 3A 6163

3300 CD 3F 3B 3A 2B 40 3C FE : 26
3308 25 20 03 3A 2C 40 32 2B : 4B
3310 40 C3 B8 30 3A 2B 40 FE : 8E
3318 33 D2 40 33 3A 2D 40 B7 : D6
3320 CA B8 30 3D 32 2D 40 3A : C8
3328 2B 40 32 2C 40 3E 33 32 : AC
3330 2B 40 3E 19 2A 29 40 B7 : 0C
3338 CB 1D CD 3F 3B C3 B8 30 : DA
3340 3A 2B 40 3C 32 2B 40 FE : 7C
3348 3C C2 B8 30 3A 2C 40 32 : BE
3350 2B 40 11 3C 3D 83 5F 1B : F2
3358 1A 2A 29 40 B7 CB 1D CD : 19
3360 3F 3B C3 B8 30 E5 C5 D5 : A4
3368 F5 2A 30 40 B7 CB 1D 24 : 52
3370 24 CD 6A 3B FE 20 C2 9F : 15
3378 33 2C CD 6A 3B FE 20 C2 : B1
---------------------------------
SUM: 96 FE FF 1D 22 A2 19 A3 EB48

3380 9F 33 25 25 2D 3E 00 CD : 54
3388 3F 3B 24 3A 32 40 11 42 : 9D
3390 3D 83 5F 1B 1A CD 3F 3B : 9B
3398 7C 32 31 40 C3 5A 34 CD : 3D
33A0 B8 3B FE BE DA 2A 34 3A : 21
33A8 32 40 FE 07 D2 2A 34 2A : D1
33B0 30 40 B7 CB 1D 3A 32 40 : BB
33B8 FE 05 38 04 2C 2C 18 01 : B0
33C0 2D 24 CD 6A 3B FE AD C2 : 30
33C8 F4 33 08 3A 32 40 FE 05 : DE
33D0 38 03 2C 18 01 2D 25 CD : 9F
33D8 6A 3B FE D6 CA 2A 34 FE : 9F
33E0 AD CA 2A 34 3A 32 40 FE : 7F
33E8 05 38 03 2D 18 01 2C 24 : D6
33F0 08 C3 12 34 FE D6 C2 2A : D1
```

```
33F8 34 25 CD 6A 3B FE D6 CA : 69
---------------------------------
SUM: 60 62 CF DF F4 FB 3E 64 78AF

3400 2A 34 FE 20 20 07 24 CD : 94
3408 6A 3B C3 12 34 FE AD C2 : 1B
3410 2A 34 FE AD 20 04 3E 20 : 8B
3418 18 02 3E AD CD 8C 3B 3A : D3
3420 32 40 32 33 40 3E 07 32 : 8E
3428 32 40 3A 32 40 FE 20 D2 : 0E
3430 BA 35 FE 07 D2 42 35 CD : 0A
3438 B8 3B E6 70 B7 C2 49 34 : 3F
3440 3A 35 40 3C E6 03 32 35 : 3B
3448 40 3A 35 40 FE 02 20 06 : 15
3450 CD 5F 34 C3 5A 34 B7 CC : 34
3458 BF 34 F1 D1 C1 E1 C9 2A : 4A
3460 30 40 B7 CB 1D 3E 00 CD : 1A
3468 3F 3B 3A 30 40 3C FE 98 : F6
3470 D2 A9 34 CB 47 28 06 32 : 21
3478 30 40 C3 A9 34 6F B7 CB : 01
---------------------------------
SUM: 23 FB CF E7 21 00 7C 81 912B

3480 1D 2C 3A 31 40 67 CD 6A : 92
3488 3B FE D6 CA A9 34 24 CD : A7
3490 6A 3B FE D6 CA A9 34 FE : 1E
3498 AD 20 07 3A 31 40 3D 32 : EE
34A0 31 40 3A 30 40 3C 32 30 : B9
34A8 40 3A 32 40 FE 05 30 02 : 21
34B0 3E 04 3C FE 07 20 02 3E : E3
34B8 05 32 32 40 C3 2E 35 2A : F9
34C0 30 40 B7 CB 1D 3E 00 CD : 1A
34C8 3F 3B 3A 30 40 3D FE 02 : 61
34D0 DA 2E 35 47 3A 29 40 C6 : ED
34D8 05 B8 D2 2E 35 78 CB 47 : 7C
34E0 20 06 32 30 40 C3 13 35 : D3
34E8 6F B7 CB 1D 3A 31 40 67 : 20
34F0 CD 6A 3B FE D6 CA 2E 35 : 73
34F8 24 CD 6A 3B FE D6 CA 2E : 62
---------------------------------
SUM: F1 8A 89 AF 06 C3 4F DC 263A

3500 35 FE AD 20 07 3A 31 40 : B2
3508 3D 32 31 40 3A 30 40 3D : C7
3510 32 30 40 3A 32 40 FE 05 : 51
3518 38 0A 3A 30 40 CB 87 32 : 70
3520 30 40 3E 00 3C FE 05 20 : 0D
3528 02 3E 01 32 32 40 21 42 : 48
3530 3D 3A 32 40 85 6F 2D 7E : 88
3538 2A 30 40 B7 CB 1D CD 3F : 45
3540 3B C9 3A 32 40 FE 09 C2 : 79
3548 9D 35 21 8F 40 11 07 00 : DA
3550 06 10 7E B7 CA 5D 35 19 : C0
3558 10 F8 C3 5A 34 3E 01 77 : 0F
3560 23 3A 30 40 B7 1F 77 23 : 3D
3568 3A 31 40 B7 17 77 23 3E : 51
3570 FF 77 23 3E 00 77 23 77 : E8
3578 23 3A 29 40 47 3A 30 40 : B7
---------------------------------
SUM: E2 74 61 3A 04 30 49 3D A87D

3580 90 B7 1F B7 1F 47 3A 2A : E7
3588 40 4F 3A 31 40 91 4F 78 : 92
3590 81 FE 80 38 02 3E 81 B7 : AF
3598 20 02 3E 01 77 2A 30 40 : 72
35A0 B7 CB 1D 3A 32 40 C6 08 : 19
35A8 CD 3F 3B D6 07 FE 0A 20 : 4C
35B0 03 3A 33 40 32 32 40 C3 : 17
35B8 5A 34 2A 30 40 B7 CB 1D : C7
35C0 3A 32 40 B7 1F C6 06 CD : 1B
35C8 3F 3B 3A 32 40 3C FE 25 : 85
35D0 20 03 3A 33 40 32 32 40 : 74
35D8 C3 5A 34 11 41 40 06 4C : 35
35E0 3E 00 12 13 10 FC 21 01 : 91
```

```
35E8 0C CD 1E 20 CD E2 1F 43 : 28
35F0 48 4F 4F 53 45 20 20 28 : E6
35F8 31 29 52 45 47 55 4C 41 : 1A
---------------------------------
SUM: 71 8D 85 99 CC 2E FD CC 7787

3600 52 20 6F 72 20 28 32 29 : F6
3608 43 4F 4E 53 54 52 55 43 : 71
3610 54 49 56 45 00 CD 21 20 : 46
3618 FE 1B CA FA 1F FE 31 DA : 05
3620 15 36 FE 33 D2 15 36 FE : 97
3628 32 CA 24 37 3E 00 32 40 : 07
3630 40 21 01 0C CD 1E 20 CD : 46
3638 E2 1F 50 55 53 48 20 30 : 91
3640 2D 39 20 54 4F 20 45 4C : DA
3648 45 43 54 20 20 5A 20 54 : EA
3650 4F 20 43 48 41 4E 47 45 : 15
3658 20 53 49 44 45 20 20 00 : 85
3660 21 25 0C CD 1E 20 3A 40 : D7
3668 40 FE 01 20 04 3E 76 18 : 2F
3670 02 3E 5E CD F4 1F CD 21 : 6C
3678 20 FE 1B CA FA 1F FE 5A : 74
---------------------------------
SUM: B4 61 D6 53 C8 44 C8 59 F257

3680 20 09 3A 40 40 3C E6 01 : 06
3688 32 40 40 FE 0D CA 52 38 : 11
3690 FE 30 DA 60 36 FE 3A D2 : A8
3698 60 36 21 6B 3F 11 13 00 : 85
36A0 D6 30 B7 CA AA 36 47 19 : C7
36A8 10 FD 54 5D 3A 40 40 4F : C7
36B0 B7 20 05 21 01 0A 18 03 : 23
36B8 21 4C 0A D9 B7 20 05 11 : 3D
36C0 42 40 18 03 11 8D 40 D9 : 54
36C8 06 13 1A D9 67 3E 00 CB : 7C
36D0 14 17 CB 14 17 CB 14 17 : 17
36D8 CB 14 17 12 47 D9 79 D9 : 7A
36E0 B7 20 03 13 18 01 1B 78 : 99
36E8 D9 CD 1A 38 26 0A 2D CB : 20
36F0 41 20 02 2C 2C D9 3E 00 : D2
36F8 CB 14 17 CB 14 17 CB 14 : CB
---------------------------------
SUM: 31 E7 D9 6E B2 1F 47 72 8ABF

3700 17 CB 14 17 12 47 D9 79 : B8
3708 D9 B7 20 03 13 18 01 1B : FA
3710 78 D9 CD 1A 38 26 0A 2D : CD
3718 CB 41 20 02 2C 2C 13 10 : A9
3720 A9 C3 60 36 3E 01 32 39 : AC
3728 40 21 01 0A CD 6A 3B 32 : 10
3730 3A 40 21 01 0C CD 1E 20 : B3
3738 CD E2 1F 34 2C 36 2D 43 : D4
3740 55 52 53 4F 52 20 4D 4F : 57
3748 56 45 20 20 20 38 2D 42 : A2
3750 55 49 4C 44 20 20 32 2D : CD
3758 44 49 4D 49 4E 49 53 48 : 55
3760 00 21 26 0C CD 1E 20 CD : 2B
3768 21 20 FE 1B CA FA 1F FE : 3B
3770 0D CA 46 38 D6 30 FE 09 : 62
3778 D2 61 37 CB 47 C2 61 37 : D6
---------------------------------
SUM: 67 37 6F D1 60 EA 4C B0 59E6

3780 FE 02 CA D2 37 FE 08 CA : A3
3788 D2 37 FE 06 CA 9F 37 3A : E7
3790 39 40 FE 01 CA 17 38 3E : CF
3798 FF 32 3B 40 C3 AC 37 3A : 8C
37A0 39 40 FE 4C CA 17 38 3E : 1A
37A8 01 32 3B 40 3A 39 40 26 : 87
37B0 0A 6F 3A 3A 40 CD 8C 3B : C1
37B8 3A 39 40 47 3A 3B 40 80 : 2F
37C0 32 39 40 6F CD 6A 3B 32 : BE
37C8 3A 40 3E 2B CD 8C 3B C3 : 3A
37D0 17 38 FE 08 CA F2 37 11 : 59
37D8 41 40 3A 39 40 26 0A 6F : D3
37E0 83 30 01 14 5F 1A B7 CA : C2
37E8 17 38 3D 12 CD 1A 38 C3 : 80
37F0 0B 38 11 41 40 3A 39 40 : 88
37F8 26 0A 6F 83 30 01 14 5F : C6
---------------------------------
SUM: 15 60 28 EB 4C 35 E5 3C D2B9

3800 1A FE 0F CA 17 38 3C CD : 49
3808 1A 38 12 26 0A 3A 39 40 : 47
3810 6F CD 6A 3B 32 3A 40 C3 : 50
3818 61 37 C5 F5 47 B7 CB 18 : 33
3820 4F FE 02 DA 2E 38 3E D6 : A3
3828 CD 8C 3B 25 10 FA CB 41 : CF
3830 CA 39 38 3E AD CD 8C 3B : BA
3838 25 3E 20 CD 8C 3B 25 BC : F8
3840 C2 3B 38 F1 C1 C9 3A 39 : 23
3848 40 6F 26 0A 3A 3A 40 CD : 60
3850 8C 3B 21 01 0C CD 1E 20 : 00
3858 CD E2 1F 20 20 20 20 20 : 6E
3860 20 20 20 20 20 20 20 20 : 00
3868 20 20 20 4F 4B 3F 20 20 : 79
3870 20 20 20 20 20 20 20 20 : 00
3878 20 20 20 20 20 20 20 20 : 00
---------------------------------
SUM: EA 82 03 F5 E3 2C 72 BC 16F6

3880 00 CD 21 20 FE 1B CA FA : EB
3888 1F FE 4E CA E6 35 C9 CD : E6
3890 5D 3C 21 01 0C CD 1E 20 : D2
3898 11 4D 3D CD E8 1F 11 74 : F4
38A0 3D 21 09 02 1A CD 8C 3B : 17
38A8 2C 13 7D FE 4D 20 03 2E : 58
38B0 09 24 3E 07 BC C2 A4 38 : CC
38B8 21 09 17 CD 1E 20 11 C8 : 25
38C0 3E CD E8 1F 26 0A CD 1E : 2D
38C8 20 CD E8 1F 11 E0 3E 25 : 48
38D0 CD 1E 20 CD E8 1F 26 16 : 1B
38D8 CD 1E 20 CD E8 1F 3E 03 : 20
38E0 32 29 40 3E 00 32 2A 40 : 75
38E8 3E 09 32 31 40 3E 97 32 : F1
38F0 30 40 3E BE 32 A3 33 CD : 41
38F8 65 33 CD AF 39 01 00 08 : 56
---------------------------------
```

```
SUM: 1D 30 35 40 CB 47 69 67 1424

3900 CD 9D 3C CD AF 39 CD 9D : C5
3908 3C CD D0 1F B7 CA F7 38 : A8
3910 C9 3E 00 32 2A 40 32 31 : 06
3918 40 32 2F 40 32 36 40 32 : BB
3920 2D 40 3E 02 32 2B 40 32 : 7C
3928 32 40 3E 03 32 29 40 3E : 8C
3930 97 32 30 40 21 01 0C CD : 34
3938 1E 20 CD E2 1F 20 20 20 : 6C
3940 20 20 20 20 20 20 20 20 : 00
3948 20 4C 45 56 45 4C 20 28 : E0
3950 31 20 2D 20 35 29 20 20 : 3C
3958 20 20 20 20 20 20 20 20 : 00
3960 20 20 20 00 CD 21 20 FE : 6C
3968 1B CA FA 1F FE 36 D2 64 : 68
3970 39 FE 31 DA 64 39 D6 31 : E6
3978 11 48 3D 83 5F 1A 32 A3 : 67
---------------------------------
SUM: 3C 88 EE B7 AE 4D 5C 53 8CF2

3980 33 CD 1E 20 CD E2 1F 53 : 5F
3988 54 4F 43 4B 3A 20 20 20 : CB
3990 20 20 20 4D 55 44 20 42 : A8
3998 41 4C 4C 49 4E 27 20 20 : D7
39A0 20 20 53 43 4F 52 45 3A : F6
39A8 30 20 2D 20 30 00 C9 E5 : 7B
39B0 C5 D5 F5 3E 00 32 38 40 : 77
39B8 D9 16 10 D9 DD 21 8F 40 : A5
39C0 DD 7E 00 B7 CA C8 3A DD : BB
39C8 6E 01 DD 66 02 B7 CB 1C : 52
39D0 3E 20 CD 8C 3B DD 66 02 : 37
39D8 DD 7E 06 DD 46 04 E6 7F : ED
39E0 32 2D 3B 80 DD 46 06 FE : 41
39E8 14 30 06 DD 77 04 C3 09 : 6E
39F0 3A D6 14 4F 3E 7F B8 38 : 20
39F8 03 25 18 01 24 79 FE 14 : F0
---------------------------------
SUM: BF 28 6F AE 09 B4 24 41 5C84

3A00 D2 F1 39 DD 74 02 DD 77 : A3
3A08 04 3E 7F B8 38 03 05 18 : D1
3A10 01 04 3E 00 B8 20 02 06 : 23
3A18 81 DD 70 06 DD 7E 03 DD : 0F
3A20 86 01 DD 77 01 DD 6E 01 : 28
3A28 DD 66 02 B7 CB 1C 3E 4C : 6D
3A30 BD CA 49 3A CD 6A 3B FE : 7A
3A38 7B CA 49 3A FE D6 CA 49 : AF
3A40 3A FE AD CA 49 3A C3 63 : 58
3A48 3A 1E 00 DD 73 00 FE 7B : 21
3A50 CA C8 3A FE D6 20 04 3E : 02
3A58 AD 18 02 3E 20 CD 8C 3B : B9
3A60 C3 C8 3A 3A 2B 40 FE 33 : 9B
3A68 D2 B1 3A DD 6E 01 DD 66 : 4C
3A70 02 EB 2A 29 40 B7 CB 1D : 1F
3A78 B7 CB 1A 7A 94 FE 02 D2 : 7C
---------------------------------
SUM: 2C 36 78 DA F7 F9 91 E5 AFBE

3A80 93 3A 7B 95 FE 02 D2 93 : 42
3A88 3A 3E 01 32 38 40 3E 00 : 61
3A90 DD 77 00 2A 30 40 B7 CB : 70
3A98 1D 7A 94 FE 02 D2 B1 3A : E8
3AA0 7B 95 FE 02 D2 B1 3A 3E : 0B
3AA8 02 32 38 40 3E 00 DD 77 : 3E
3AB0 00 DD 6E 01 DD 66 02 7C : 0D
3AB8 CB 47 20 04 3E DF 18 02 : 6D
3AC0 3E A1 B7 CB 1C CD 8C 3B : 11
3AC8 01 07 00 DD 09 D9 15 7A : 56
3AD0 D9 FE 00 C2 C0 39 3A 38 : 04
3AD8 40 FE 01 C2 FF 3A 3A 2B : 9F
3AE0 40 FE 06 30 03 32 2C 40 : 15
3AE8 3E 20 32 2B 40 21 26 0C : 4E
3AF0 CD 1E 20 3A 2F 40 3C 32 : 22
3AF8 2F 40 C6 30 CD F4 1F 3A : 7F
---------------------------------
SUM: E1 74 AA 27 B6 EA 6B 9B B404

3B00 38 40 FE 02 C2 28 3B 3A : D7
3B08 32 40 FE 06 30 03 32 33 : 0E
3B10 40 3E 20 32 32 40 21 22 : 85
3B18 0C CD 1E 20 3A 36 40 3C : 03
3B20 32 36 40 C6 30 CD F4 1F : 7E
3B28 F1 D1 C1 E1 C9 00 3A 2F : 96
3B30 40 FE 08 CA E5 3B 3A 36 : A0
3B38 40 FE 08 CA EF 3B C9 E5 : E8
3B40 C5 D5 F5 EB 26 00 6F 29 : 38
3B48 29 01 D4 3C 09 EB 1A CD : 15
3B50 8C 3B 2C 13 1A CD 8C 3B : B4
3B58 2D 24 13 1A CD 8C 3B 2C : 3E
3B60 13 1A CD 8C 3B F1 D1 C1 : 44
3B68 E1 C9 E5 7C FE 0C 38 04 : 51
3B70 3E 20 E1 C9 B7 20 04 3E : 21
3B78 20 E1 C9 7D FE 27 38 07 : AB
---------------------------------
SUM: 52 A7 AF 37 2F 6C 94 9B F37B

3B80 D6 26 6F 7C C6 0D 67 CD : EE
3B88 1B 20 E1 C9 E5 F5 7C FE : 39
3B90 0B 38 03 F1 E1 C9 B7 20 : B8
3B98 03 F1 E1 C9 7D FE 4D 38 : 9E
3BA0 03 F1 E1 C9 FE 27 38 07 : 02
3BA8 D6 26 6F 7C C6 0D 67 CD : EE
3BB0 1E 20 F1 CD F4 1F E1 C9 : B9
3BB8 C5 D5 E5 21 3C 40 CD DD : C6
3BC0 3B F5 D1 21 3D 40 CD DD : 49
3BC8 3B F5 C1 79 AB 1F 2A 3E : 9C
3BD0 40 ED 6A 22 3E 40 E1 D1 : E9
3BD8 C1 3A 3E 40 C9 46 2A 3E : F0
3BE0 40 29 10 FD C9 CD 35 3C : 7D
3BE8 D1 21 2F 02 C3 F6 3B CD : E4
3BF0 35 3C D1 21 09 02 11 F8 : 77
3BF8 3E 01 00 00 1A CD 8C 3B : ED
---------------------------------
SUM: B6 13 A4 4E 9B D3 43 03 A8C3

3C00 13 2C 04 3E 17 B8 20 08 : 78
3C08 06 00 0C 7D D6 17 6F 24 : 0F
```

```
3C10 3E 05 B9 C2 FC 3B 01 00 : F6
3C18 FF CD 9D 3C 21 01 0C CD : A0
3C20 1E 20 11 4D 3D CD E8 1F : AD
3C28 CD C4 1F CD D0 1F B7 CA : ED
3C30 2B 3C C3 07 30 06 19 C5 : 45
3C38 CD 3D 30 CD 65 33 CD AF : 1B
3C40 39 01 00 40 CD 9D 3C C1 : E1
3C48 10 ED C9 3E 28 CD 30 20 : 49
3C50 C9 21 8F 40 06 70 3E 00 : 6D
3C58 77 23 10 FC C9 3E 0C CD : 86
3C60 F4 1F 06 18 21 27 00 CD : 46
3C68 1E 20 3E 7B CD F4 1F CD : A4
3C70 F4 1F 24 10 F2 11 AB 3C : 31
3C78 21 00 00 CD 1E 20 CD E8 : E1
---------------------------------
SUM: E9 EB 59 D1 6E 94 6E C2 82AF

3C80 1F 21 00 0B CD 1E 20 CD : 23
3C88 E8 1F 21 00 0D CD 1E 20 : 40
3C90 CD E8 1F 21 00 18 CD 1E : F8
3C98 20 CD E8 1F C9 C5 00 00 : 82
3CA0 00 00 3E 00 0B B8 C2 9E : 61
3CA8 3C C1 C9 7B 7B 7B 7B 7B : 2D
3CB0 7B 7B 7B 7B 7B 7B 7B 7B : D8
3CB8 7B 7B 7B 7B 7B 7B 7B 7B : D8
3CC0 7B 7B 7B 7B 7B 7B 7B 7B : D8
3CC8 7B 7B 7B 7B 7B 7B 7B 7B : D8
3CD0 7B 7B 0D 0D 20 20 20 20 : 90
3CD8 3C A8 2D 29 20 28 20 C7 : 69
3CE0 28 DA 2F 49 2F A8 29 29 : A3
3CE8 2F A8 2F 28 2F A8 C9 49 : 17
3CF0 2D A8 C9 49 20 2F 2F 29 : 8E
3CF8 20 2D 2F 29 2D 3E 28 CD : 05
---------------------------------
SUM: 77 1C AB CB 00 EC BD 5F A9B6

3D00 29 20 C4 20 2F 29 49 2D : FB
3D08 2F A8 28 28 2F A8 2F 29 : 56
3D10 2C CD 49 29 2C 2D 28 28 : 14
3D18 CD 20 2F 28 2D 20 2F 28 : E8
3D20 CD 20 20 29 2D 3C 20 20 : DF
3D28 20 20 2D 3D 20 C9 28 20 : DB
3D30 3E 2D 20 20 20 20 3D 2D : 55
3D38 20 20 2F 68 01 02 03 02 : DF
3D40 04 05 0B 0A 0B 0C 0D 0E : 50
3D48 DC BE 96 64 28 20 20 20 : 1C
3D50 20 20 20 20 20 20 20 20 : 00
3D58 20 20 48 49 54 20 41 4E : D4
3D60 59 20 4B 45 59 20 20 20 : C2
3D68 20 20 20 20 20 20 20 20 : 00
3D70 20 20 20 0D D6 D6 20 20 : 59
3D78 20 D6 D6 20 D6 D6 20 20 : D8
---------------------------------
SUM: 75 7B 6A F0 F1 9D 65 31 2E07

3D80 20 D6 D6 20 D6 D6 D6 D6 : 44
3D88 D6 D6 20 20 20 20 20 20 : 6C
3D90 20 20 D6 D6 D6 D6 D6 20 : 8E
3D98 20 20 D6 D6 D6 D6 20 20 : D8
3DA0 D6 D6 20 20 20 20 D6 D6 : D8
3DA8 20 20 20 20 D6 D6 20 D6 : 22
3DB0 D6 20 20 D6 D6 20 D6 D6 : 8E
3DB8 D6 D6 D6 20 D6 D6 D6 20 : 44
3DC0 D6 D6 20 20 20 D6 D6 20 : D8
3DC8 D6 D6 20 20 20 D6 D6 20 : D8
3DD0 20 20 20 20 20 20 D6 D6 : 6C
3DD8 20 20 D6 D6 20 D6 D6 20 : D8
3DE0 20 D6 D6 20 D6 D6 20 20 : D8
3DE8 20 20 D6 D6 20 20 20 20 : 6C
3DF0 D6 D6 20 D6 D6 D6 20 D6 : 44
3DF8 D6 20 20 D6 D6 D6 20 D6 : 8E
---------------------------------
SUM: B0 B0 FA FA 66 D2 66 FA 98A8

3E00 20 D6 D6 20 D6 D6 20 20 : D8
3E08 20 D6 D6 20 D6 D6 20 20 : D8
3E10 20 D6 D6 20 20 20 20 20 : 6C
3E18 20 20 D6 D6 D6 D6 D6 20 : 8E
3E20 20 D6 D6 20 20 D6 D6 20 : D8
3E28 D6 D6 20 20 20 20 D6 D6 : D8
3E30 20 20 20 20 D6 D6 20 D6 : 22
3E38 D6 D6 D6 D6 D6 20 D6 20 : 44
3E40 D6 D6 20 20 20 D6 D6 20 : D8
3E48 D6 D6 20 20 20 D6 D6 20 : D8
3E50 D6 D6 20 20 20 D6 D6 20 : D8
3E58 20 20 20 20 20 20 D6 D6 : 6C
3E60 20 20 D6 D6 20 D6 D6 D6 : 8E
3E68 D6 D6 D6 20 D6 D6 20 20 : 8E
3E70 20 20 D6 D6 20 20 20 20 : 6C
3E78 D6 D6 20 D6 D6 20 D6 D6 : 44
---------------------------------
SUM: FA D2 66 8E FA 1C 1C 8E 1D69

3E80 D6 20 20 20 D6 D6 20 20 : 22
3E88 20 D6 D6 20 20 D6 D6 D6 : 8E
3E90 D6 D6 20 20 D6 D6 D6 D6 : 44
3E98 D6 D6 20 20 20 20 20 20 : 6C
3EA0 20 20 D6 D6 D6 D6 D6 20 : 8E
3EA8 20 D6 D6 20 20 D6 D6 20 : D8
3EB0 D6 D6 D6 D6 D6 20 D6 D6 : FA
3EB8 D6 D6 D6 20 D6 D6 20 D6 : 44
3EC0 D6 20 20 D6 D6 20 20 20 : 22
3EC8 D6 D6 D6 D6 D6 D6 D6 D6 : B0
3ED0 D6 D6 D6 D6 D6 D6 D6 D6 : B0
3ED8 D6 D6 D6 D6 D6 D6 D6 0D : E7
3EE0 AD D6 D6 D6 D6 D6 D6 D6 : 87
3EE8 D6 D6 D6 D6 D6 D6 D6 D6 : B0
3EF0 D6 D6 D6 D6 D6 D6 AD 0D : BE
3EF8 D6 D6 20 20 20 D6 D6 20 : D8
---------------------------------
SUM: 15 3E D2 66 88 3E 5F 8A F120

3F00 D6 D6 20 D6 D6 20 20 20 : D8
3F08 D6 D6 20 20 20 D6 D6 D6 : 8E
3F10 D6 20 D6 20 D6 D6 20 D6 : 8E
3F18 D6 20 D6 D6 D6 20 20 D6 : 8E
3F20 D6 20 20 D6 D6 D6 D6 D6 : 44
3F28 20 D6 20 D6 D6 20 D6 D6 : 8E
```

▶グラフィック関係のスタッフが必要なんでしょ。力を貸したいけど，うでがないので……。非常に残念である。数年後，お会いしましょう。　丸山　淳平(16)神奈川県

```
3F30 20 D6 D6 D6 D6 D6 D6 D6 : FA
3F38 20 20 D6 D6 20 D6 D6 D6 : 8E
3F40 D6 D6 D6 D6 20 D6 D6 20 : 44
3F48 D6 D6 20 20 D6 D6 D6 20 : 8E
3F50 20 20 20 20 20 D6 D6 20 : 6C
3F58 D6 D6 20 20 D6 D6 20 D6 : 8E
3F60 D6 20 20 20 D6 D6 20 D6 : D8
3F68 D6 20 20 00 00 00 00 00 : 16
3F70 00 00 00 00 00 00 00 00 : 00
3F78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: DC BA 4E 9A 06 BC 50 06 B732

3F80 00 12 34 56 78 9A BC DE : 48
3F88 FF ED CB A9 87 65 43 21 : B0
3F90 00 00 00 12 23 45 67 89 : 6A
3F98 AB CD EF FF 00 00 00 00 : 66
3FA0 00 00 00 00 32 10 01 23 : 66
3FA8 FF EE DC BA 98 76 54 32 : 17
3FB0 10 00 00 00 00 00 00 00 : 10
3FB8 01 22 2E EE EE EE EE 00 : 09
3FC0 00 00 00 00 00 00 00 00 : 00
3FC8 00 00 00 01 23 45 67 89 : 59
3FD0 AB 00 44 44 55 43 21 10 : FC
3FD8 00 00 00 00 00 66 66 66 : 32
```

```
3FE0 66 66 66 66 66 66 66 66 : 30
3FE8 66 00 00 00 00 00 00 00 : 66
3FF0 88 66 88 66 88 66 88 66 : B8
3FF8 88 66 88 66 88 53 10 00 : C7
---------------------------------
SUM: 41 0E B2 2F C8 C5 95 A8 CF66

4000 00 00 00 FF DB 97 53 10 : D4
4008 00 EE EE EE EE EE EE EE : 82
4010 00 00 00 00 00 00 FF DB : DA
4018 97 53 31 00 01 33 57 9B : 41
4020 DF F0 00 00 00 00 00 00 : CF
4028 00 03 09 02 00 00 00 00 : 0E
4030 97 09 02 00 00 00 00 00 : A2
4038 00 00 00 00 10 02 14 56 : 7C
4040 00 00 00 00 00 00 00 00 : 00
4048 00 00 00 00 00 00 00 00 : 00
4050 00 00 00 00 00 00 00 00 : 00
4058 00 00 00 00 00 00 00 00 : 00
4060 00 00 00 00 00 00 00 00 : 00
4068 00 00 00 00 00 00 00 00 : 00
4070 00 00 00 00 00 00 00 00 : 00
4078 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 0D 3D 2A EF DA BA AB CA EB80
```

```
4080 00 00 00 00 00 00 00 00 : 00
4088 00 00 00 00 00 00 00 00 : 00
4090 00 00 00 00 00 00 00 00 : 00
4098 00 00 00 00 00 00 00 00 : 00
40A0 00 00 00 00 00 00 00 00 : 00
40A8 00 00 00 00 00 00 00 00 : 00
40B0 00 00 00 00 00 00 00 00 : 00
40B8 00 00 00 00 00 00 00 00 : 00
40C0 00 00 00 00 00 00 00 00 : 00
40C8 00 00 00 00 00 00 00 00 : 00
40D0 00 00 00 00 00 00 00 00 : 00
40D8 00 00 00 00 00 00 00 00 : 00
40E0 00 00 00 00 00 00 00 00 : 00
40E8 00 00 00 00 00 00 00 00 : 00
40F0 00 00 00 00 00 00 00 00 : 00
40F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 00 00 00 00 00 00 00 0000

4100 00                      : 00
---------------------------------
SUM: 00 00 00 00 00 00 00 00 0000
```

リスト2

```
0000              1  ; ---------------------------
0000              2  ;  MUD BALLIN'
0000              3  ;          PROGRAMMED BY  A.S
0000              4  ; ---------------------------
0000              5  OFFSET $9000
3000              6  START  $3000
3000              7
3000              8 @PRINT  EQU  $1FF4
3000              9 @SPACE  EQU  $1FF1
3000             10 @MSG    EQU  $1FE8
3000             11 @GETKY  EQU  $1FD0
3000             12 @INKEY  EQU  $2021
3000             13 @BELL   EQU  $1FC4
3000             14 @WIDCH  EQU  $2030
3000             15 @LOC    EQU  $201E
3000             16 @SCRN   EQU  $201B
3000             17 @PRNTA  EQU  $1FC1
3000             18 @PAUSE  EQU  $1FC7
3000             19 @MPRNT  EQU  $1FE2
3000             20 @HOT    EQU  $1FFA
3000             21
3000 CD 4B 3C    22  CALL   #INIT
3003 ED 73 FF    23  LD     (SPBUF),SP
3006 40
3007             24 #HOTSTART
3007 ED 7B FF    25  LD     SP,(SPBUF)
300A 40
300B CD 51 3C    26  CALL   #BALLINIT
300E CD 8F 38    27  CALL   #OPEN
3011 CD 5D 3C    28  CALL   #SCREEN
3014 CD DB 35    29  CALL   #PILE
3017 CD 11 39    30  CALL   #INIT2
301A CD 51 3C    31  CALL   #BALLINIT
301D             32 #LOOP
301D CD 3D 30    33  CALL   #PLAYER
3020 CD 65 33    34  CALL   #ENEMY
3023 CD 2E 3B    35  CALL   #CHECK
3026 CD AF 39    36  CALL   #BALL
3029 01 00 08    37  LD     BC,$0800
302C CD 9D 3C    38  CALL   #WAIT
302F CD AF 39    39  CALL   #BALL
3032 CD 9D 3C    40  CALL   #WAIT
3035 C3 1D 30    41  JP     #LOOP
3038             42
3038             43 #QUIT:
3038 ED 7B FF   .44  LD     SP,(SPBUF)
303B 40
303C C9          45  RET
303D             46
303D             47 #PLAYER
303D 21 07 0C    48  LD     HL,$0C07
3040 CD 1E 20    49  CALL   @LOC
3043 3A 2D 40    50  LD     A,(#STOCK)
3046 06 20       51  LD     B," "
3048 B7 28 02    52  IF     A<>0   THEN LD B,"o"
304B 06 6F
304D 78          53  LD     A,B
304E CD F4 1F    54  CALL   @PRINT
3051 3A 29 40    55  LD     A,(#PLX)
3054 B7          56  OR     A
3055 1F          57  RRA
3056 6F          58  LD     L,A
3057 3A 2A 40    59  LD     A,(#PLY)
305A 67          60  LD     H,A
305B 24          61  INC    H
305C 24          62  INC    H
305D CD 6A 3B    63  CALL   #SCRN
3060 FE 20 C2    64  IF     A<>" " JP #PLSTEP1
3063 81 30
3065 2C          65  INC    L
3066 CD 6A 3B    66  CALL   #SCRN
3069 FE 20 C2    67  IF     A<>" " JP #PLSTEP1
306C 81 30
306E 2A 29 40    68  LD     HL,(#PLX)
3071 B7          69  OR     A
3072 CB 1D       70  RR     L
3074 3E 00       71  LD     A,0
3076 CD 3F 3B    72  CALL   #PUT
3079 24          73  INC    H
307A 7C          74  LD     A,H
307B 32 2A 40    75  LD     (#PLY),A
307E C3 4A 32    76  JP     #PLSTEP2
3081             77 #PLSTEP1
3081 3A 2B 40    78  LD     A,(#PLFM)
3084 FE 32 D2    79  IF     A>=50  JP #CROUCH
3087 14 33
3089 FE 20 D2    80  IF     A>=32  JP #DOWN
308C F3 32
308E FE 07 D2    81  IF     A>=7   JP #SHOOT
3091 5E 32
3093             82 #PLRET1
3093 CD D0 1F    83  CALL   @GETKY
3096 F5          84  PUSH   AF
3097 08          85  EX     AF,AF'
3098 F1          86  POP    AF
3099 08          87  EX     AF,AF'
309A FE 32 CA    88  IF     A="2"  JP #CROUCH
309D 14 33
309F FE 34 CC    89  IF     A="4"  CALL #LEFTMOVE
30A2 76 31
30A4 FE 36 CC    90  IF     A="6"  CALL #RIGHTMOVE
30A7 DA 31
30A9 FE 47 CC    91  IF     A="G"  CALL #HOTSTART
30AC 07 30
30AE FE 5A CC    92  IF     A="Z"  CALL #DIG
30B1 C2 30
30B3 FE 58 CC    93  IF     A="X"  CALL #SHOOT
30B6 5E 32
30B8             94 #PLRET2
30B8 08          95  EX     AF,AF'
30B9 32 37 40    96  LD     (#LASTKY),A
30BC CD C7 1F    97  CALL   @PAUSE
30BF 38 30       98  DW     #QUIT
30C1 C9          99  RET
30C2            100 #DIG
30C2 3A 37 40   101  LD     A,(#LASTKY)
30C5 FE 5A C8   102  IF     A="Z"  RET
30C8 3A 2D 40   103  LD     A,(#STOCK)
30CB FE 01 CA   104  IF     A=1    JP #DROP
30CE 1F 31
30D0 3A 29 40   105  LD     A,(#PLX)
30D3 B7         106  OR     A
30D4 1F         107  RRA
30D5 6F         108  LD     L,A
30D6 3A 2A 40   109  LD     A,(#PLY)
30D9 67         110  LD     H,A
30DA 3A 2B 40   111  LD     A,(#PLFM)
30DD FE 05 38   112  IF     A>=5   THEN DEC L ELSE INC L : INC L
30E0 03 2D 18
30E3 02 2C 2C
30E6 24         113  INC    H
30E7 CD 6A 3B   114  CALL   #SCRN
30EA FE AD CA   115  IF     A="ｭ"  JP #PLDSTEP1'
30ED 01 31
30EF FE 20 C8   116  IF     A=" "  RET
30F2 25         117  DEC    H
30F3 CD 6A 3B   118  CALL   #SCRN
30F6 FE D6 C8   119  IF     A="ﾖ"  RET
30F9 FE 20 20   120  IF     A=" "  THEN INC H : CALL #SCRN
30FC 04 24 CD
30FF 6A 3B
3101            121 #PLDSTEP1'
3101 FE D6 CA   122  IF     A="ﾖ"  JP #PLDSTEP1
3104 0C 31
3106 FE AD CA   123  IF     A="ｭ"  JP #PLDSTEP2
3109 14 31
310B C9         124  RET
310C            125 #PLDSTEP1
310C 3E AD      126  LD     A,"ｭ"
310E CD 8C 3B   127  CALL   #PRINT
3111 C3 19 31   128  JP     #PLDSTEP3
3114            129 #PLDSTEP2
3114 3E 20      130  LD     A," "
3116 CD 8C 3B   131  CALL   #PRINT
3119            132 #PLDSTEP3
3119 3E 01      133  LD     A,1
311B 32 2D 40   134  LD     (#STOCK),A
311E C9         135  RET
311F            136 #DROP
311F 3E 00      137  LD     A,0
3121 32 2D 40   138  LD     (#STOCK),A
3124 3A 29 40   139  LD     A,(#PLX)
3127 B7         140  OR     A
3128 1F         141  RRA
3129 6F         142  LD     L,A
312A 3A 2A 40   143  LD     A,(#PLY)
312D 67         144  LD     H,A
312E 3A 2B 40   145  LD     A,(#PLFM)
3131 FE 05 38   146  IF     A>=5   THEN DEC L ELSE INC L : INC L
3134 03 2D 18
3137 02 2C 2C
313A 24         147  INC    H
313B CD 6A 3B   148  CALL   #SCRN
313E FE AD CA   149  IF     A="ｭ"  JP #PLOSTEP3
3141 70 31
3143 FE 20 CA   150  IF     A=" "  JP #PLOSTEP1
3146 57 31
3148 25         151  DEC    H
3149 CD 6A 3B   152  CALL   #SCRN
314C FE 20 CA   153  IF     A=" "  JP #PLOSTEP2
314F 65 31
3151 3E 01      154  LD     A,1
3153 32 2D 40   155  LD     (#STOCK),A
3156 C9         156  RET
3157            157 #PLOSTEP1
3157 24         158  INC    H
3158 CD 6A 3B   159  CALL   #SCRN
315B FE 20 CA   160  IF     A=" "  JP #PLOSTEP1
315E 57 31
3160 FE AD 28   161  IF     A<>"ｭ" THEN DEC H
3163 01 25
3165            162 #PLOSTEP2
3165 FE AD CA   163  IF     A="ｭ"  JP #PLOSTEP3
3168 70 31
316A 3E AD      164  LD     A,"ｭ"
316C CD 8C 3B   165  CALL   #PRINT
316F C9         166  RET
3170            167 #PLOSTEP3
3170 3E D6      168  LD     A,"ﾖ"
3172 CD 8C 3B   169  CALL   #PRINT
3175 C9         170  RET
3176            171 #LEFTMOVE
3176 2A 29 40   172  LD     HL,(#PLX)
3179 B7         173  OR     A
317A CB 1D      174  RR     L
317C 3E 00      175  LD     A,0
317E CD 3F 3B   176  CALL   #PUT
3181 3A 29 40   177  LD     A,(#PLX)
3184 3D         178  DEC    A
3185 FE 98 D2   179  IF     A>=152 JP #PLLSTEP1
3188 C4 31
318A FE 02 DA   180  IF     A<2    JP #PLLSTEP1
318D C4 31
318F CB 47      181  BIT    0,A
```

▶今年の成人式に出ようと思っていたのに……。ライトスタッフの日本ゲームコンテストの締め切りが１月16日なので，１日中部屋にこもってCGを描いてしまった。なんて年だ。
藤原　常雅(20)神奈川県

```
3191 20 06 32   182  IF     Z        THEN LD (#PLX),A : JP #PLLSTEP1
3194 29 40 C3
3197 C4 31
3199 6F         183  LD     L,A
319A B7         184  OR     A
319B CB 1D      185  RR     L
319D 3A 2A 40   186  LD     A,(#PLY)
31A0 67         187  LD     H,A
31A1 CD 6A 3B   188  CALL   #SCRN
31A4 FE D6 CA   189  IF     A="ﾖ"  JP #PLLSTEP1
31A7 C4 31
31A9 24         190  INC    H
31AA CD 6A 3B   191  CALL   #SCRN
31AD FE D6 CA   192  IF     A="ﾖ"  JP #PLLSTEP1
31B0 C4 31
31B2 FE AD 20   193  IF     A="ｭ"  THEN LD A,(#PLY) : DEC A : LD (#PLY),A
31B5 07 3A 2A
31B8 40 3D 32
31BB 2A 40
31BD 3A 29 40   194  LD     A,(#PLX)
31C0 3D         195  DEC    A
31C1 32 29 40   196  LD     (#PLX),A
31C4            197 #PLLSTEP1
31C4 3A 2B 40   198  LD     A,(#PLFM)
31C7 FE 05 30   199  IF     A<5    THEN LD A,4
31CA 02 3E 04
31CD 3C         200  INC    A
31CE FE 07 20   201  IF     A=7    THEN LD A,5
31D1 02 3E 05
31D4 32 2B 40   202  LD     (#PLFM),A
31D7 C3 4A 32   203  JP     #PLSTEP2
31DA            204 #RIGHTMOVE
31DA 2A 29 40   205  LD     HL,(#PLX)
31DD B7         206  OR     A
31DE CB 1D      207  RR     L
31E0 3E 00      208  LD     A,0
31E2 CD 3F 3B   209  CALL   #PUT
31E5 3A 29 40   210  LD     A,(#PLX)
31E8 3C         211  INC    A
31E9 FE 98 D2   212  IF     A>=152 JP #PLSTEP2
31EC 4A 32
31EE 47         213  LD     B,A
31EF 3A 30 40   214  LD     A,(#ENX)
31F2 D6 05      215  SUB    5
31F4 B8 DA 4A   216  IF     A<B    JP #PLSTEP2
31F7 32
31F8 78         217  LD     A,B
31F9 CB 47      218  BIT    0,A
31FB 28 06 32   219  IF     NZ       THEN LD (#PLX),A : JP #PLRSTEP1
31FE 29 40 C3
3201 2F 32
3203 6F         220  LD     L,A
3204 B7         221  OR     A
3205 CB 1D      222  RR     L
3207 3A 2A 40   223  LD     A,(#PLY)
320A 67         224  LD     H,A
320B 2C         225  INC    L
320C CD 6A 3B   226  CALL   #SCRN
320F FE D6 CA   227  IF     A="ﾖ"  JP #PLSTEP2
3212 4A 32
3214 24         228  INC    H
3215 CD 6A 3B   229  CALL   #SCRN
3218 FE D6 CA   230  IF     A="ﾖ"  JP #PLSTEP2
321B 4A 32
321D FE AD 20   231  IF     A="ｭ"  THEN LD A,(#PLY) : DEC A : LD (#PLY),A
3220 07 3A 2A
3223 40 3D 32
3226 2A 40
3228 3A 29 40   232  LD     A,(#PLX)
322B 3C         233  INC    A
322C 32 29 40   234  LD     (#PLX),A
322F            235 #PLRSTEP1
322F 3A 2B 40   236  LD     A,(#PLFM)
3232 FE 05 38   237  IF     A>=5   THEN LD A,(#PLX) : RES 0,A : LD (#PLX),A : LD
 A,0
3235 0A 3A 29
3238 40 CB 87
323B 32 29 40
323E 3E 00
3240 3C         238  INC    A
3241 FE 05 20   239  IF     A=5    THEN LD A,1
3244 02 3E 01
3247 32 2B 40   240  LD     (#PLFM),A
324A            241 #PLSTEP2
324A 21 3C 3D   242  LD     HL,#PLFMTAB
324D 3A 2B 40   243  LD     A,(#PLFM)
3250 85         244  ADD    A,L
3251 6F         245  LD     L,A
3252 2D         246  DEC    L
3253 7E         247  LD     A,(HL)
3254 2A 29 40   248  LD     HL,(#PLX)
3257 B7         249  OR     A
3258 CB 1D      250  RR     L
325A CD 3F 3B   251  CALL   #PUT
325D C9         252  RET
325E            253 #SHOOT
325E 08         254  EX     AF,AF'
325F 3E 58      255  LD     A,"X"
3261 08         256  EX     AF,AF'
3262 3A 2B 40   257  LD     A,(#PLFM)
3265 FE 0E D2   258  IF     A>=14 JP #PLSSTEP1
3268 89 32
326A 3A 37 40   259  LD     A,(#LASTKY)
326D FE 58 C8   260  IF     A="X" RET
3270 3A 2D 40   261  LD     A,(#STOCK)
3273 B7 C8      262  IF     A=0    RET
3275 3D         263  DEC    A
3276 32 2D 40   264  LD     (#STOCK),A
3279 3A 2B 40   265  LD     A,(#PLFM)
327C 32 2C 40   266  LD     (#PLFMS),A
327F 3E 0D      267  LD     A,13
3281 32 2B 40   268  LD     (#PLFM),A
3284 3E 00      269  LD     A,0
3286 32 2E 40   270  LD     (#PLAIM),A
3289            271 #PLSSTEP1
3289 2A 29 40   272  LD     HL,(#PLX)
328C B7         273  OR     A
328D CB 1D      274  RR     L
328F B7         275  OR     A
3290 1F         276  RRA
3291 CD 3F 3B   277  CALL   #PUT
3294 CD D0 1F   278  CALL   @GETKY
3297 FE 58 20   279  IF     A="X"  THEN LD A,(#PLAIM) : ADD A,6 : LD (#PLAIM),A
329A 08 3A 2E
329D 40 C6 06
32A0 32 2E 40
32A3 3A 2B 40   280  LD     A,(#PLFM)
32A6 3C         281  INC    A
32A7 FE 12 CA   282  IF     A=18    JP #PLSSTEP2
32AA B9 32
32AC FE 14 20   283  IF     A=20    THEN LD A,(#PLFMS)
32AF 03 3A 2C
32B2 40
32B3 32 2B 40   284  LD     (#PLFM),A
32B6 C3 F0 32   285  JP     #PLSSTEP3
32B9            286 #PLSSTEP2
32B9 21 8F 40   287  LD     HL,#MUDBALL
32BC 01 07 00   288  LD     BC,7
32BF            289 #PLTLOOP
32BF 7E         290  LD     A,(HL)
32C0 B7 28 04   291  IF     A<>0    THEN ADD HL,BC : JP #PLTLOOP
32C3 09 C3 BF
32C6 32
32C7 3E 01      292  LD     A,1
32C9 77         293  LD     (HL),A
32CA 23         294  INC    HL
32CB 3A 29 40   295  LD     A,(#PLX)
32CE B7         296  OR     A
32CF CB 1F      297  RR     A
32D1 3C         298  INC    A
32D2 77         299  LD     (HL),A
32D3 23         300  INC    HL
32D4 3A 2A 40   301  LD     A,(#PLY)
32D7 B7         302  OR     A
32D8 17         303  RLA
32D9 77         304  LD     (HL),A
32DA 23         305  INC    HL
32DB 3E 01      306  LD     A,1
32DD 77         307  LD     (HL),A
32DE 23         308  INC    HL
32DF 3E 00      309  LD     A,0
32E1 77         310  LD     (HL),A
32E2 23         311  INC    HL
32E3 77         312  LD     (HL),A
32E4            313  ; DELTA-Y' <--- FOR AIMING
32E4 3A 2E 40   314  LD     A,(#PLAIM)
32E7 D6 05      315  SUB    5
32E9 23         316  INC    HL
32EA 77         317  LD     (HL),A
32EB 3E 12      318  LD     A,18
32ED 32 2B 40   319  LD     (#PLFM),A
32F0            320 #PLSSTEP3
32F0 C3 B8 30   321  JP     #PLRET2
32F3            322 #DOWN
32F3 2A 29 40   323  LD     HL,(#PLX)
32F6 B7         324  OR     A
32F7 CB 1D      325  RR     L
32F9 3A 2B 40   326  LD     A,(#PLFM)
32FC B7         327  OR     A
32FD 1F         328  RRA
32FE C6 03      329  ADD    A,3
3300 CD 3F 3B   330  CALL   #PUT
3303 3A 2B 40   331  LD     A,(#PLFM)
3306 3C         332  INC    A
3307 FE 25 20   333  IF     A=$25  THEN LD A,(#PLFMS)
330A 03 3A 2C
330D 40
330E 32 2B 40   334  LD     (#PLFM),A
3311 C3 B8 30   335  JP     #PLRET2
3314            336 #CROUCH
3314 3A 2B 40   337  LD     A,(#PLFM)
3317 FE 33 D2   338  IF     A>=51 JP #CRSTEP1
331A 40 33
331C 3A 2D 40   339  LD     A,(#STOCK)
331F B7 CA B8   340  IF     A=0    JP #PLRET2
3322 30
3323 3D         341  DEC    A
3324 32 2D 40   342  LD     (#STOCK),A
3327 3A 2B 40   343  LD     A,(#PLFM)
332A 32 2C 40   344  LD     (#PLFMS),A
332D 3E 33      345  LD     A,51
332F 32 2B 40   346  LD     (#PLFM),A
3332 3E 19      347  LD     A,25
3334 2A 29 40   348  LD     HL,(#PLX)
3337 B7         349  OR     A
3338 CB 1D      350  RR     L
333A CD 3F 3B   351  CALL   #PUT
333D C3 B8 30   352  JP     #PLRET2
3340            353 #CRSTEP1
3340 3A 2B 40   354  LD     A,(#PLFM)
3343 3C         355  INC    A
3344 32 2B 40   356  LD     (#PLFM),A
3347 FE 3C C2   357  IF     A<>60 JP #PLRET2
334A B8 30
334C 3A 2C 40   358  LD     A,(#PLFMS)
334F 32 2B 40   359  LD     (#PLFM),A
3352 11 3C 3D   360  LD     DE,#PLFMTAB
3355 83         361  ADD    A,E
3356 5F         362  LD     E,A
3357 1B         363  DEC    DE
3358 1A         364  LD     A,(DE)
3359 2A 29 40   365  LD     HL,(#PLX)
335C B7         366  OR     A
335D CB 1D      367  RR     L
335F CD 3F 3B   368  CALL   #PUT
3362 C3 B8 30   369  JP     #PLRET2
3365            370 #ENEMY
3365 E5         371  PUSH   HL
3366 C5         372  PUSH   BC
3367 D5         373  PUSH   DE
3368 F5         374  PUSH   AF
3369 2A 30 40   375  LD     HL,(#ENX)
336C B7         376  OR     A
336D CB 1D      377  RR     L
336F 24         378  INC    H
3370 24         379  INC    H
3371 CD 6A 3B   380  CALL   #SCRN
3374 FE 20 C2   381  IF     A<>" " JP #ENSTEP1
3377 9F 33
3379 2C         382  INC    L
337A CD 6A 3B   383  CALL   #SCRN
337D FE 20 C2   384  IF     A<>" " JP #ENSTEP1
3380 9F 33
3382 25         385  DEC    H
3383 25         386  DEC    H
3384 2D         387  DEC    L
3385 3E 00      388  LD     A,0
3387 CD 3F 3B   389  CALL   #PUT
338A 24         390  INC    H
338B 3A 32 40   391  LD     A,(#ENFM)
338E 11 42 3D   392  LD     DE,#ENFMTAB
3391 83         393  ADD    A,E
3392 5F         394  LD     E,A
3393 1B         395  DEC    DE
3394 1A         396  LD     A,(DE)
3395 CD 3F 3B   397  CALL   #PUT
3398 7C         398  LD     A,H
3399 32 31 40   399  LD     (#ENY),A
339C C3 5A 34   400  JP     #ENSTEP2
339F            401 #ENSTEP1
339F CD B8 3B   402  CALL   #RND
33A2            403 #LEVEL
33A2 FE BE DA   404  IF     A<190  JP #ENSTEP1''
33A5 2A 34
33A7 3A 32 40   405  LD     A,(#ENFM)
33AA FE 07 D2   406  IF     A>=7   JP #ENSTEP1''
33AD 2A 34
33AF 2A 30 40   407  LD     HL,(#ENX)
33B2 B7         408  OR     A
33B3 CB 1D      409  RR     L
33B5 3A 32 40   410  LD     A,(#ENFM)
33B8 FE 05 38   411  IF     A>=5   THEN INC L : INC L ELSE DEC L
33BB 04 2C 2C
33BE 18 01 2D
33C1 24         412  INC    H
33C2 CD 6A 3B   413  CALL   #SCRN
33C5 FE AD C2   414  IF     A<>"ｭ" JP #ENTSTEP12
33C8 F4 33
33CA 08         415  EX     AF,AF'
33CB 3A 32 40   416  LD     A,(#ENFM)
33CE FE 05 38   417  IF     A>=5   THEN INC L ELSE DEC L
33D1 03 2C 18
33D4 01 2D
33D6 25         418  DEC    H
33D7 CD 6A 3B   419  CALL   #SCRN
33DA FE D6 CA   420  IF     A="ﾖ"  JP #ENSTEP1''
33DD 2A 34
33DF FE AD CA   421  IF     A="ｭ"  JP #ENSTEP1''
33E2 2A 34
33E4 3A 32 40   422  LD     A,(#ENFM)
33E7 FE 05 38   423  IF     A>=5   THEN DEC L ELSE INC L
33EA 03 2D 18
33ED 01 2C
33EF 24         424  INC    H
33F0 08         425  EX     AF,AF'
33F1 C3 12 34   426  JP     #ENTSTEP1
33F4            427 #ENTSTEP12
33F4 FE D6 C2   428  IF     A<>"ﾖ" JP #ENSTEP1''
33F7 2A 34
33F9 25         429  DEC    H
```

▶なに，いよいよ女の子をひいきしているだって？　ばかものっ！　女の子はひいきしていいのだ。STUDIO Xには女の子のハガキを優先して載せなさい。私が許す!

山崎　寿夫(22)千葉県

```
33FA CD 6A 3B   430  CALL   #SCRN
33FD FE D6 CA   431  IF     A="ﾖ"  JP #ENSTEP1''
3400 2A 34
3402 FE 20 20   432  IF     A=" "  THEN INC H : CALL #SCRN : JP #ENTSTEP1
3405 07 24 CD
3408 6A 3B C3
340B 12 34
340D FE AD C2   433  IF     A<>"ｭ" JP #ENSTEP1''
3410 2A 34
3412            434 #ENTSTEP1
3412 FE AD 20   435  IF     A="ｭ"  THEN LD A," " ELSE LD A,"ｭ"
3415 04 3E 20
3418 18 02 3E
341B AD
341C CD 8C 3B   436  CALL   #PRINT
341F 3A 32 40   437  LD     A,(#ENFM)
3422 32 33 40   438  LD     (#ENFMS),A
3425 3E 07      439  LD     A,7
3427 32 32 40   440  LD     (#ENFM),A
342A            441 #ENSTEP1''
342A 3A 32 40   442  LD     A,(#ENFM)
342D FE 20 D2   443  IF     A>=32  JP #DOWN2
3430 BA 35
3432 FE 07 D2   444  IF     A>=7   JP #THROW
3435 42 35
3437 CD B8 3B   445  CALL   #RND
343A E6 70      446  AND    $70
343C B7 C2 49   447  IF     A<>0   JP #ENSTEP1'
343F 34
3440 3A 35 40   448  LD     A,(#ENMOVE)
3443 3C         449  INC    A
3444 E6 03      450  AND    3
3446 32 35 40   451  LD     (#ENMOVE),A
3449            452 #ENSTEP1'
3449 3A 35 40   453  LD     A,(#ENMOVE)
344C FE 02 20   454  IF     A=2    THEN CALL #RIGHTEN : JP #ENSTEP2
344F 06 CD 5F
3452 34 C3 5A
3455 34
3456 B7 CC BF   455  IF     A=0    CALL #LEFTEN
3459 34
345A            456 #ENSTEP2
345A F1         457  POP    AF
345B D1         458  POP    DE
345C C1         459  POP    BC
345D E1         460  POP    HL
345E C9         461  RET
345F            462 #RIGHTEN
345F 2A 30 40   463  LD     HL,(#ENX)
3462 B7         464  OR     A
3463 CB 1D      465  RR     L
3465 3E 00      466  LD     A,0
3467 CD 3F 3B   467  CALL   #PUT
346A 3A 30 40   468  LD     A,(#ENX)
346D 3C         469  INC    A
346E FE 98 D2   470  IF     A>=152 JP #ENRSTEP1
3471 A9 34
3473 CB 47      471  BIT    0,A
3475 28 06 32   472  IF     NZ     THEN LD (#ENX),A : JP #ENRSTEP1
3478 30 40 C3
347B A9 34
347D 6F         473  LD     L,A
347E B7         474  OR     A
347F CB 1D      475  RR     L
3481 2C         476  INC    L
3482 3A 31 40   477  LD     A,(#ENY)
3485 67         478  LD     H,A
3486 CD 6A 3B   479  CALL   #SCRN
3489 FE D6 CA   480  IF     A="ﾖ"  JP #ENRSTEP1
348C A9 34
348E 24         481  INC    H
348F CD 6A 3B   482  CALL   #SCRN
3492 FE D6 CA   483  IF     A="ﾖ"  JP #ENRSTEP1
3495 A9 34
3497 FE AD 20   484  IF     A="ｭ"  THEN LD A,(#ENY) : DEC A : LD (#ENY),A
349A 07 3A 31
349D 40 3D 32
34A0 31 40
34A2 3A 30 40   485  LD     A,(#ENX)
34A5 3C         486  INC    A
34A6 32 30 40   487  LD     (#ENX),A
34A9            488 #ENRSTEP1
34A9 3A 32 40   489  LD     A,(#ENFM)
34AC FE 05 30   490  IF     A<5    THEN LD A,4
34AF 02 3E 04
34B2 3C         491  INC    A
34B3 FE 07 20   492  IF     A=7    THEN LD A,5
34B6 02 3E 05
34B9 32 32 40   493  LD     (#ENFM),A
34BC C3 2E 35   494  JP     #ENSTEP3
34BF            495 #LEFTEN
34BF 2A 30 40   496  LD     HL,(#ENX)
34C2 B7         497  OR     A
34C3 CB 1D      498  RR     L
34C5 3E 00      499  LD     A,0
34C7 CD 3F 3B   500  CALL   #PUT
34CA 3A 30 40   501  LD     A,(#ENX)
34CD 3D         502  DEC    A
34CE FE 02 DA   503  IF     A<2    JP #ENSTEP3
34D1 2E 35
34D3 47         504  LD     B,A
34D4 3A 29 40   505  LD     A,(#PLX)
34D7 C6 05      506  ADD    A,5
34D9 B8 D2 2E   507  IF     A>=B   JP #ENSTEP3
34DC 35
34DD 78         508  LD     A,B
34DE CB 47      509  BIT    0,A
34E0 20 06 32   510  IF     Z      THEN LD (#ENX),A : JP #ENLSTEP1
34E3 30 40 C3
34E6 13 35
34E8 6F         511  LD     L,A
34E9 B7         512  OR     A
34EA CB 1D      513  RR     L
34EC 3A 31 40   514  LD     A,(#ENY)
34EF 67         515  LD     H,A
34F0 CD 6A 3B   516  CALL   #SCRN
34F3 FE D6 CA   517  IF     A="ﾖ"  JP #ENSTEP3
34F6 2E 35
34F8 24         518  INC    H
34F9 CD 6A 3B   519  CALL   #SCRN
34FC FE D6 CA   520  IF     A="ﾖ"  JP #ENSTEP3
34FF 2E 35
3501 FE AD 20   521  IF     A="ｭ"  THEN LD A,(#ENY) : DEC A : LD (#ENY),A
3504 07 3A 31
3507 40 3D 32
350A 31 40
350C 3A 30 40   522  LD     A,(#ENX)
350F 3D         523  DEC    A
3510 32 30 40   524  LD     (#ENX),A
3513            525 #ENLSTEP1
3513 3A 32 40   526  LD     A,(#ENFM)
3516 FE 05 38   527  IF     A>=5   THEN LD A,(#ENX) : RES 0,A : LD (#ENX),A : LD
 A,0
3519 0A 3A 30
351C 40 CB 87
351F 32 30 40
3522 3E 00
3524 3C         528  INC    A
3525 FE 05 20   529  IF     A=5    THEN LD A,1
3528 02 3E 01
352B 32 32 40   530  LD     (#ENFM),A
352E            531 #ENSTEP3
352E 21 42 3D   532  LD     HL,#ENFMTAB
3531 3A 32 40   533  LD     A,(#ENFM)
3534 85         534  ADD    A,L
3535 6F         535  LD     L,A
3536 2D         536  DEC    L
3537 7E         537  LD     A,(HL)
3538 2A 30 40   538  LD     HL,(#ENX)
353B B7         539  OR     A
353C CB 1D      540  RR     L
353E CD 3F 3B   541  CALL   #PUT
3541 C9         542  RET
3542            543 #THROW
3542 3A 32 40   544  LD     A,(#ENFM)
3545 FE 09 C2   545  IF     A<>9   JP #ETSTEP2
3548 9D 35
354A 21 8F 40   546  LD     HL,#MUDBALL
354D 11 07 00   547  LD     DE,7
3550 06 10      548  LD     B,16
3552            549 #ETLOOP
3552 7E         550  LD     A,(HL)
3553 B7 CA 5D   551  IF     A=0    JP #ETSTEP1
3556 35
3557 19         552  ADD    HL,DE
3558 10 F8      553  DJNZ   #ETLOOP
355A C3 5A 34   554  JP     #ENSTEP2
355D            555 #ETSTEP1
355D 3E 01      556  LD     A,1
355F 77         557  LD     (HL),A
3560 23         558  INC    HL
3561 3A 30 40   559  LD     A,(#ENX)
3564 B7         560  OR     A
3565 1F         561  RRA
3566 77         562  LD     (HL),A
3567 23         563  INC    HL
3568 3A 31 40   564  LD     A,(#ENY)
356B B7         565  OR     A
356C 17         566  RLA
356D 77         567  LD     (HL),A
356E 23         568  INC    HL
356F 3E FF      569  LD     A,$FF
3571 77         570  LD     (HL),A
3572 23         571  INC    HL
3573 3E 00      572  LD     A,0
3575 77         573  LD     (HL),A
3576 23         574  INC    HL
3577 77         575  LD     (HL),A
3578 23         576  INC    HL
3579 3A 29 40   577  LD     A,(#PLX)
357C 47         578  LD     B,A
357D 3A 30 40   579  LD     A,(#ENX)
3580 90         580  SUB    B
3581 B7         581  OR     A
3582 1F         582  RRA
3583 B7         583  OR     A
3584 1F         584  RRA
3585 47         585  LD     B,A
3586 3A 2A 40   586  LD     A,(#PLY)
3589 4F         587  LD     C,A
358A 3A 31 40   588  LD     A,(#ENY)
358D 91         589  SUB    C
358E 4F         590  LD     C,A
358F 78         591  LD     A,B
3590 81         592  ADD    A,C
3591 FE 80 38   593  IF     A>=128 THEN LD A,$81
3594 02 3E 81
3597 B7 20 02   594  IF     A=0    THEN LD  A,1
359A 3E 01
359C 77         595  LD     (HL),A
359D            596 #ETSTEP2
359D 2A 30 40   597  LD     HL,(#ENX)
35A0 B7         598  OR     A
35A1 CB 1D      599  RR     L
35A3 3A 32 40   600  LD     A,(#ENFM)
35A6 C6 08      601  ADD    A,8
35A8 CD 3F 3B   602  CALL   #PUT
35AB D6 07      603  SUB    7
35AD FE 0A 20   604  IF     A=10   THEN LD A,(#ENFMS)
35B0 03 3A 33
35B3 40
35B4 32 32 40   605  LD     (#ENFM),A
35B7 C3 5A 34   606  JP     #ENSTEP2
35BA            607 #DOWN2
35BA 2A 30 40   608  LD     HL,(#ENX)
35BD B7         609  OR     A
35BE CB 1D      610  RR     L
35C0 3A 32 40   611  LD     A,(#ENFM)
35C3 B7         612  OR     A
35C4 1F         613  RRA
35C5 C6 06      614  ADD    A,6
35C7 CD 3F 3B   615  CALL   #PUT
35CA 3A 32 40   616  LD     A,(#ENFM)
35CD 3C         617  INC    A
35CE FE 25 20   618  IF     A=$25  THEN LD A,(#ENFMS)
35D1 03 3A 33
35D4 40
35D5 32 32 40   619  LD     (#ENFM),A
35D8 C3 5A 34   620  JP     #ENSTEP2
35DB            621 #PILE
35DB 11 41 40   622  LD     DE,#CONST
35DE 06 4C      623  LD     B,76
35E0 3E 00      624  LD     A,0
35E2            625 #PILOOP0
35E2 12         626  LD     (DE),A
35E3 13         627  INC    DE
35E4 10 FC      628  DJNZ   #PILOOP0
35E6            629 #REIN0
35E6 21 01 0C   630  LD     HL,$0C01
35E9 CD 1E 20   631  CALL   @LOC
35EC CD E2 1F   632  CALL   @MPRNT
35EF 43 48 4F   633  DM     "CHOOSE  (1)REGULAR or (2)CONSTRUCTIVE"
35F2 4F 53 45
35F5 20 20 28
35F8 31 29 52
35FB 45 47 55
35FE 4C 41 52
3601 20 6F 72
3604 20 28 32
3607 29 43 4F
360A 4E 53 54
360D 52 55 43
3610 54 49 56
3613 45
3614 00         634  DB     0
3615            635 #REIN1
3615 CD 21 20   636  CALL   @INKEY
3618 FE 1B CA   637  IF     A=1BH  JP @HOT
361B FA 1F
361D FE 31 DA   638  IF     A<"1"  JP #REIN1
3620 15 36
3622 FE 33 D2   639  IF     A>="3" JP #REIN1
3625 15 36
3627 FE 32 CA   640  IF     A="2"  JP #CONSTRUCTIVE
362A 24 37
362C            641  ; ' REGULAR
362C 3E 00      642  LD     A,0
362E 32 40 40   643  LD     (#SIDE),A
3631 21 01 0C   644  LD     HL,$0C01
3634 CD 1E 20   645  CALL   @LOC
3637 CD E2 1F   646  CALL   @MPRNT
363A 50 55 53   647  DM     "PUSH 0-9 TO ELECT 'Z TO CHANGE SIDE  "
363D 48 20 30
3640 2D 39 20
3643 54 4F 20
3646 45 4C 45
3649 43 54 20
364C 20 5A 20
364F 54 4F 20
3652 43 48 41
3655 4E 47 45
3658 20 53 49
365B 44 45 20
365E 20
365F 00         648  DB     0
3660            649 #REIN2
3660 21 25 0C   650  LD     HL,$0C00+37
3663 CD 1E 20   651  CALL   @LOC
3666 3A 40 40   652  LD     A,(#SIDE)
3669 FE 01 20   653  IF     A=1    THEN LD A,"v" ELSE LD A,"^"
```

▶ある日の物語。ギャラガ'88をやろうと思い，ディスクを入れロードした。だが，ロードが終わっても，画面には星が点々と出ているだけで動かない。「なぜだ。まさか壊れたのか?!」と思い調べてみると，キーボードカバーがESCキーにひっかかっていて，ポーズがかかっていただけだった。　西川　敏弘(20)神奈川県

```
366C 04 3E 76
366F 18 02 3E
3672 5E
3673 CD F4 1F   654  CALL   @PRINT
3676 CD 21 20   655  CALL   @INKEY
3679 FE 1B CA   656  IF     A=1BH  JP @HOT
367C FA 1F
367E FE 5A 20   657  IF     A="Z"  THEN LD A,(#SIDE) : INC A : AND 1 : LD (#SIDE
),A
3681 09 3A 40
3684 40 3C E6
3687 01 32 40
368A 40
368B FE 0D CA   658  IF     A=$D   JP #OK?
368E 52 38
3690 FE 30 DA   659  IF     A<"0"  JP #REIN2
3693 60 36
3695 FE 3A D2   660  IF     A>=":" JP #REIN2
3698 60 36
369A 21 6B 3F   661  LD     HL,#GRFORM
369D 11 13 00   662  LD     DE,19
36A0 D6 30      663  SUB    "0"
36A2 B7 CA AA   664  IF     A=0    JP #RESTEP1
36A5 36
36A6 47         665  LD     B,A
36A7            666 #RELOOP1
36A7 19         667  ADD    HL,DE
36A8 10 FD      668  DJNZ   #RELOOP1
36AA            669 #RESTEP1
36AA 54 5D      670  LD     DE,HL
36AC 3A 40 40   671  LD     A,(#SIDE)
36AF 4F         672  LD     C,A
36B0 B7 20 05   673  IF     A=0    THEN LD  HL,$0A01 ELSE LD HL,$0A00+76
36B3 21 01 0A
36B6 18 03 21
36B9 4C 0A
36BB D9         674  EXX
36BC B7 20 05   675  IF     A=0    THEN LD  DE,#CONST+1 ELSE LD DE,#CONST+76
36BF 11 42 40
36C2 18 03 11
36C5 8D 40
36C7 D9         676  EXX
36C8 06 13      677  LD     B,19
36CA            678 #RELOOP2
36CA 1A         679  LD     A,(DE)
36CB D9         680  EXX
36CC 67         681  LD     H,A
36CD 3E 00      682  LD     A,0
36CF CB 14      683  RL     H
36D1 17         684  RLA
36D2 CB 14      685  RL     H
36D4 17         686  RLA
36D5 CB 14      687  RL     H
36D7 17         688  RLA
36D8 CB 14      689  RL     H
36DA 17         690  RLA
36DB 12         691  LD     (DE),A
36DC 47         692  LD     B,A
36DD D9         693  EXX
36DE 79         694  LD     A,C
36DF D9         695  EXX
36E0 B7 20 03   696  IF     A=0    THEN INC DE ELSE DEC DE
36E3 13 18 01
36E6 1B
36E7 78         697  LD     A,B
36E8 D9         698  EXX
36E9 CD 1A 38   699  CALL   #PILEUP
36EC 26 0A      700  LD     H,10
36EE 2D         701  DEC    L
36EF CB 41      702  BIT    0,C
36F1 20 02 2C   703  IF     Z      THEN INC L : INC L
36F4 2C
36F5 D9         704  EXX
36F6 3E 00      705  LD     A,0
36F8 CB 14      706  RL     H
36FA 17         707  RLA
36FB CB 14      708  RL     H
36FD 17         709  RLA
36FE CB 14      710  RL     H
3700 17         711  RLA
3701 CB 14      712  RL     H
3703 17         713  RLA
3704 12         714  LD     (DE),A
3705 47         715  LD     B,A
3706 D9         716  EXX
3707 79         717  LD     A,C
3708 D9         718  EXX
3709 B7 20 03   719  IF     A=0    THEN INC DE ELSE DEC DE
370C 13 18 01
370F 1B
3710 78         720  LD     A,B
3711 D9         721  EXX
3712 CD 1A 38   722  CALL   #PILEUP
3715 26 0A      723  LD     H,10
3717 2D         724  DEC    L
3718 CB 41      725  BIT    0,C
371A 20 02 2C   726  IF     Z      THEN INC L : INC L
371D 2C
371E 13         727  INC    DE
371F 10 A9      728  DJNZ   #RELOOP2
3721 C3 60 36   729  JP     #REIN2
3724            730 #CONSTRUCTIVE
3724 3E 01      731  LD     A,1
3726 32 39 40   732  LD     (#CUX),A
3729 21 01 0A   733  LD     HL,$0A01
372C CD 6A 3B   734  CALL   #SCRN
372F 32 3A 40   735  LD     (#CHCUR),A
3732 21 01 0C   736  LD     HL,$0C01
3735 CD 1E 20   737  CALL   @LOC
3738 CD E2 1F   738  CALL   @MPRNT
373B 34 2C 36   739  DM     "4,6-CURSOR MOVE   8-BUILD  2-DIMINISH"
373E 2D 43 55
3741 52 53 4F
3744 52 20 4D
3747 4F 56 45
374A 20 20 20
374D 38 2D 42
3750 55 49 4C
3753 44 20 20
3756 32 2D 44
3759 49 4D 49
375C 4E 49 53
375F 48
3760 00         740  DB     0
3761            741 #COLOOP2
3761 21 26 0C   742  LD     HL,$0C00+38
3764 CD 1E 20   743  CALL   @LOC
3767 CD 21 20   744  CALL   @INKEY
376A FE 1B CA   745  IF     A=1BH  JP @HOT
376D FA 1F
376F FE 0D CA   746  IF     A=$0D  JP #CONRE
3772 46 38
3774 D6 30      747  SUB    "0"
3776 FE 09 D2   748  IF     A>=9   JP #COLOOP2
3779 61 37
377B CB 47      749  BIT    0,A
377D C2 61 37   750  IF     NZ     JP #COLOOP2
3780 FE 02 CA   751  IF     A=2    JP #BUILD
3783 D2 37
3785 FE 08 CA   752  IF     A=8    JP #BUILD
3788 D2 37
378A FE 06 CA   753  IF     A=6    JP #COSTEP1
378D 9F 37
378F 3A 39 40   754  LD     A,(#CUX)
3792 FE 01 CA   755  IF     A=1    JP #COSTEP2
3795 17 38
3797 3E FF      756  LD     A,255
3799 32 3B 40   757  LD     (#MOVE),A
379C C3 AC 37   758  JP     #COSTEP1'
379F            759 #COSTEP1
379F 3A 39 40   760  LD     A,(#CUX)
37A2 FE 4C CA   761  IF     A=76   JP #COSTEP2
37A5 17 38
37A7 3E 01      762  LD     A,1
37A9 32 3B 40   763  LD     (#MOVE),A
37AC            764 #COSTEP1'
37AC 3A 39 40   765  LD     A,(#CUX)
37AF 26 0A      766  LD     H,10
37B1 6F         767  LD     L,A
37B2 3A 3A 40   768  LD     A,(#CHCUR)
37B5 CD 8C 3B   769  CALL   #PRINT
37B8 3A 39 40   770  LD     A,(#CUX)
37BB 47         771  LD     B,A
37BC 3A 3B 40   772  LD     A,(#MOVE)
37BF 80         773  ADD    A,B
37C0 32 39 40   774  LD     (#CUX),A
37C3 6F         775  LD     L,A
37C4 CD 6A 3B   776  CALL   #SCRN
37C7 32 3A 40   777  LD     (#CHCUR),A
37CA 3E 2B      778  LD     A,"+"
37CC CD 8C 3B   779  CALL   #PRINT
37CF C3 17 38   780  JP     #COSTEP2
37D2            781 #BUILD
37D2 FE 08 CA   782  IF     A=8    JP #BUSTEP1
37D5 F2 37
37D7 11 41 40   783  LD     DE,#CONST
37DA 3A 39 40   784  LD     A,(#CUX)
37DD 26 0A      785  LD     H,10
37DF 6F         786  LD     L,A
37E0 83         787  ADD    A,E
37E1 30 01 14   788  IF     C      THEN INC D
37E4 5F         789  LD     E,A
37E5 1A         790  LD     A,(DE)
37E6 B7 CA 17   791  IF     A=0    JP #COSTEP2
37E9 38
37EA 3D         792  DEC    A
37EB 12         793  LD     (DE),A
37EC CD 1A 38   794  CALL   #PILEUP
37EF C3 0B 38   795  JP     #BUSTEP2
37F2            796 #BUSTEP1
37F2 11 41 40   797  LD     DE,#CONST
37F5 3A 39 40   798  LD     A,(#CUX)
37F8 26 0A      799  LD     H,10
37FA 6F         800  LD     L,A
37FB 83         801  ADD    A,E
37FC 30 01 14   802  IF     C      THEN INC D
37FF 5F         803  LD     E,A
3800 1A         804  LD     A,(DE)
3801 FE 0F CA   805  IF     A=15   JP #COSTEP2
3804 17 38
3806 3C         806  INC    A
3807 CD 1A 38   807  CALL   #PILEUP
380A 12         808  LD     (DE),A
380B            809 #BUSTEP2
380B 26 0A      810  LD     H,10
380D 3A 39 40   811  LD     A,(#CUX)
3810 6F         812  LD     L,A
3811 CD 6A 3B   813  CALL   #SCRN
3814 32 3A 40   814  LD     (#CHCUR),A
3817            815 #COSTEP2
3817 C3 61 37   816  JP     #COLOOP2
381A            817 #PILEUP
381A            818  ; A <-- HEIGHT
381A C5         819  PUSH   BC
381B F5         820  PUSH   AF
381C 47         821  LD     B,A
381D B7         822  OR     A
381E CB 18      823  RR     B
3820 4F         824  LD     C,A
3821 FE 02 DA   825  IF     A<2    JP #FISTEP1
3824 2E 38
3826 3E D6      826  LD     A,"ﾖ"
3828            827 #FILOOP1
3828 CD 8C 3B   828  CALL   #PRINT
382B 25         829  DEC    H
382C 10 FA      830  DJNZ   #FILOOP1
382E            831 #FISTEP1
382E CB 41      832  BIT    0,C
3830 CA 39 38   833  IF     Z      JP #FISTEP2
3833 3E AD      834  LD     A,"ｭ"
3835 CD 8C 3B   835  CALL   #PRINT
3838 25         836  DEC    H
3839            837 #FISTEP2
3839 3E 20      838  LD     A," "
383B            839 #FILOOP2
383B CD 8C 3B   840  CALL   #PRINT
383E 25         841  DEC    H
383F BC C2 3B   842  IF     A<>H   JP #FILOOP2
3842 38
3843 F1         843  POP    AF
3844 C1         844  POP    BC
3845 C9         845  RET
3846            846 #CONRE
3846 3A 39 40   847  LD     A,(#CUX)
3849 6F         848  LD     L,A
384A 26 0A      849  LD     H,10
384C 3A 3A 40   850  LD     A,(#CHCUR)
384F CD 8C 3B   851  CALL   #PRINT
3852            852 #REIN3
3852            853 #OK?
3852 21 01 0C   854  LD     HL,$0C01
3855 CD 1E 20   855  CALL   @LOC
3858 CD E2 1F   856  CALL   @MPRNT
385B 20 20 20   857  DM     "                 OK?
385E 20 20 20
3861 20 20 20
3864 20 20 20
3867 20 20 20
386A 20 4F 4B
386D 3F 20 20
3870 20 20 20
3873 20 20 20
3876 20 20 20
3879 20 20 20
387C 20 20 20
387F 20
3880 00         858  DB     0
3881 CD 21 20   859  CALL   @INKEY
3884 FE 1B CA   860  IF     A=1BH  JP @HOT
3887 FA 1F
3889 FE 4E CA   861  IF     A="N"  JP #REIN0
388C E6 35
388E C9         862  RET
388F            863 #OPEN
388F CD 5D 3C   864  CALL   #SCREEN
3892 21 01 0C   865  LD     HL,$0C01
3895 CD 1E 20   866  CALL   @LOC
3898 11 4D 3D   867  LD     DE,#ANYKEY
389B CD E8 1F   868  CALL   @MSG
389E 11 74 3D   869  LD     DE.#TITLE
38A1 21 09 02   870  LD     HL,$0209
38A4            871 #LOOPOP
38A4 1A         872  LD     A,(DE)
38A5 CD 8C 3B   873  CALL   #PRINT
38A8 2C         874  INC    L
38A9 13         875  INC    DE
38AA 7D         876  LD     A,L
38AB FE 4D 20   877  IF     A=77   THEN LD L,9 : INC H
38AE 03 2E 09
38B1 24
38B2 3E 07      878  LD     A,7
38B4 BC C2 A4   879  IF     A<>H   JP #LOOPOP
38B7 38
38B8 21 09 17   880  LD     HL,$1709
38BB CD 1E 20   881  CALL   @LOC
38BE 11 C8 3E   882  LD     DE,#DEMOMUD
38C1 CD E8 1F   883  CALL   @MSG
38C4 26 0A      884  LD     H,10
38C6 CD 1E 20   885  CALL   @LOC
38C9 CD E8 1F   886  CALL   @MSG
```

▶都会に住んでいる人にはスーパーファミコンは買いにくいでしょうが，僕の住んでいるところは田舎で，なんと発売直後からスーパーファミコンがあまっています。

本田　和正(16)愛媛県

```
38CC 11 E0 3E   887  LD     DE,#DEMOMUD'
38CF 25         888  DEC    H
38D0 CD 1E 20   889  CALL   @LOC
38D3 CD E8 1F   890  CALL   @MSG
38D6 26 16      891  LD     H,22
38D8 CD 1E 20   892  CALL   @LOC
38DB CD E8 1F   893  CALL   @MSG
38DE 3E 03      894  LD     A,3
38E0 32 29 40   895  LD     (#PLX),A
38E3 3E 00      896  LD     A,0
38E5 32 2A 40   897  LD     (#PLY),A
38E8 3E 09      898  LD     A,9
38EA 32 31 40   899  LD     (#ENY),A
38ED 3E 97      900  LD     A,151
38EF 32 30 40   901  LD     (#ENX),A
38F2 3E BE      902  LD     A,190
38F4 32 A3 33   903  LD     (#LEVEL+1),A
38F7            904 #LOOPOP'
38F7 CD 65 33   905  CALL   #ENEMY
38FA CD AF 39   906  CALL   #BALL
38FD 01 00 08   907  LD     BC,$0800
3900 CD 9D 3C   908  CALL   #WAIT
3903 CD AF 39   909  CALL   #BALL
3906 CD 9D 3C   910  CALL   #WAIT
3909 CD D0 1F   911  CALL   @GETKY
390C B7 CA F7   912  IF     A=0    JP #LOOP0P'
390F 38
3910 C9         913  RET
3911            914 #INIT2
3911 3E 00      915  LD     A,0
3913 32 2A 40   916  LD     (#PLY),A
3916 32 31 40   917  LD     (#ENY),A
3919 32 2F 40   918  LD     (#PLDM),A
391C 32 36 40   919  LD     (#ENDM),A
391F 32 2D 40   920  LD     (#STOCK),A
3922 3E 02      921  LD     A,2
3924 32 2B 40   922  LD     (#PLFM),A
3927 32 32 40   923  LD     (#ENFM),A
392A 3E 03      924  LD     A,3
392C 32 29 40   925  LD     (#PLX),A
392F 3E 97      926  LD     A,151
3931 32 30 40   927  LD     (#ENX),A
3934 21 01 0C   928  LD     HL,$0C01
3937 CD 1E 20   929  CALL   @LOC
393A CD E2 1F   930  CALL   @MPRNT
393D 20 20 20   931  DM     "             LEVEL (1 - 5)            "
3940 20 20 20
3943 20 20 20
3946 20 20 20
3949 4C 45 56
394C 45 4C 20
394F 28 31 20
3952 2D 20 35
3955 29 20 20
3958 20 20 20
395B 20 20 20
395E 20 20 20
3961 20 20
3963 00         932  DB     0
3964            933 #I2LOOP
3964 CD 21 20   934  CALL   @INKEY
3967 FE 1B CA   935  IF     A=1BH  JP @HOT
396A FA 1F
396C FE 36 D2   936  IF     A>="6" JP #I2LOOP
396F 64 39
3971 FE 31 DA   937  IF     A<"1"  JP #I2LOOP
3974 64 39
3976 D6 31      938  SUB    "1"
3978 11 48 3D   939  LD     DE,#DIFICULTY
397B 83         940  ADD    A,E
397C 5F         941  LD     E,A
397D 1A         942  LD     A,(DE)
397E 32 A3 33   943  LD     (#LEVEL+1),A
3981 CD 1E 20   944  CALL   @LOC
3984 CD E2 1F   945  CALL   @MPRNT
3987 53 54 4F   946  DM     "STOCK:      MUD BALLIN"
398A 43 4B 3A
398D 20 20 20
3990 20 20 20
3993 4D 55 44
3996 20 42 41
3999 4C 4C 49
399C 4E
399D 27         947  DB     27H                          ;37H marks " ' "
399E 20 20 20   948  DM     "    SCORE:0 - 0"
39A1 20 53 43
39A4 4F 52 45
39A7 3A 30 20
39AA 2D 20 30
39AD 00         949  DB     0
39AE C9         950  RET
39AF            951 #BALL
39AF E5         952  PUSH   HL
39B0 C5         953  PUSH   BC
39B1 D5         954  PUSH   DE
39B2 F5         955  PUSH   AF
39B3 3E 00      956  LD     A,0
39B5 32 38 40   957  LD     (#HIT),A
39B8 D9         958  EXX
39B9 16 10      959  LD     D,16
39BB D9         960  EXX
39BC DD 21 8F   961  LD     IX,#MUDBALL
39BF 40
39C0            962 #LOOPBA
39C0 DD 7E 00   963  LD     A,(IX)
39C3 B7 CA C8   964  IF     A=0    JP #BASTEP
39C6 3A
39C7 DD 6E 01   965  LD     L,(IX+1)
39CA DD 66 02   966  LD     H,(IX+2)
39CD B7         967  OR     A
39CE CB 1C      968  RR     H
39D0 3E 20      969  LD     A," "
39D2 CD 8C 3B   970  CALL   #PRINT
39D5 DD 66 02   971  LD     H,(IX+2)
39D8 DD 7E 06   972  LD     A,(IX+6)
39DB DD 46 04   973  LD     B,(IX+4)
39DE E6 7F      974  AND    $7F
39E0 32 2D 3B   975  LD     (#SSS),A
39E3 80         976  ADD    A,B
39E4 DD 46 06   977  LD     B,(IX+6)
39E7 FE 14 30   978  IF     A<20   THEN LD (IX+4),A : JP #BASTEP1
39EA 06 DD 77
39ED 04 C3 09
39F0 3A
39F1            979 #BALOOP2
39F1 D6 14      980  SUB    20
39F3 4F         981  LD     C,A
39F4 3E 7F      982  LD     A,127
39F6 B8 38 03   983  IF     A>=B   THEN DEC H  ELSE INC H
39F9 25 18 01
39FC 24
39FD 79         984  LD     A,C
39FE FE 14 D2   985  IF     A>=20  JP #BALOOP2
3A01 F1 39
3A03 DD 74 02   986  LD     (IX+2),H
3A06 DD 77 04   987  LD     (IX+4),A
3A09            998 #BASTEP1
3A09 3E 7F      989  LD     A,127
3A0B B8 38 03   990  IF     A>=B   THEN DEC B  ELSE INC B
3A0E 05 18 01
3A11 04
3A12 3E 00      991  LD     A,0
3A14 B8 20 02   992  IF     A=B    THEN LD B,129
3A17 06 81
3A19 DD 70 06   993  LD     (IX+6),B
3A1C DD 7E 03   994  LD     A,(IX+3)
3A1F DD 86 01   995  ADD    A,(IX+1)
3A22 DD 77 01   996  LD     (IX+1),A
3A25 DD 6E 01   997  LD     L,(IX+1)
3A28 DD 66 02   998  LD     H,(IX+2)
3A2B B7          999  OR     A
3A2C CB 1C      1000  RR     H
3A2E 3E 4C      1001  LD     A,76
3A30 BD CA 49   1002  IF     A=L    JP #BASTEP2'
3A33 3A
3A34 CD 6A 3B   1003  CALL   #SCRN
3A37 FE 7B CA   1004  IF     A="{"  JP #BASTEP2'
3A3A 49 3A
3A3C FE D6 CA   1005  IF     A="ᄇ"  JP #BASTEP2'
3A3F 49 3A
3A41 FE AD CA   1006  IF     A="ュ"  JP #BASTEP2'
3A44 49 3A
3A46 C3 63 3A   1007  JP     #BASTEP2
3A49            1008 #BASTEP2'
3A49 1E 00      1009  LD     E,0
3A4B DD 73 00   1010  LD     (IX),E
3A4E FE 7B CA   1011  IF     A="{"  JP #BASTEP
3A51 C8 3A
3A53 FE D6 20   1012  IF     A="ᄇ"  THEN LD A,"ュ" ELSE LD A," "
3A56 04 3E AD
3A59 18 02 3E
3A5C 20
3A5D CD 8C 3B   1013  CALL   #PRINT
3A60 C3 C8 3A   1014  JP     #BASTEP
3A63            1015 #BASTEP2
3A63 3A 2B 40   1016  LD     A,(#PLFM)
3A66 FE 33 D2   1017  IF     A>=51  JP #CHSTEP2
3A69 B1 3A
3A6B DD 6E 01   1018  LD     L,(IX+1)
3A6E DD 66 02   1019  LD     H,(IX+2)
3A71 EB         1020  EX     HL,DE
3A72 2A 29 40   1021  LD     HL,(#PLX)
3A75 B7         1022  OR     A
3A76 CB 1D      1023  RR     L
3A78 B7         1024  OR     A
3A79 CB 1A      1025  RR     D
3A7B 7A         1026  LD     A,D
3A7C 94         1027  SUB    H
3A7D FE 02 D2   1028  IF     A>=2   JP #CHSTEP1
3A80 93 3A
3A82 7B         1029  LD     A,E
3A83 95         1030  SUB    L
3A84 FE 02 D2   1031  IF     A>=2   JP #CHSTEP1
3A87 93 3A
3A89 3E 01      1032  LD     A,1
3A8B 32 38 40   1033  LD     (#HIT),A
3A8E 3E 00      1034  LD     A,0
3A90 DD 77 00   1035  LD     (IX),A
3A93            1036 #CHSTEP1
3A93 2A 30 40   1037  LD     HL,(#ENX)
3A96 B7         1038  OR     A
3A97 CB 1D      1039  RR     L
3A99 7A         1040  LD     A,D
3A9A 94         1041  SUB    H
3A9B FE 02 D2   1042  IF     A>=2   JP #CHSTEP2
3A9E B1 3A
3AA0 7B         1043  LD     A,E
3AA1 95         1044  SUB    L
3AA2 FE 02 D2   1045  IF     A>=2   JP #CHSTEP2
3AA5 B1 3A
3AA7 3E 02      1046  LD     A,2
3AA9 32 38 40   1047  LD     (#HIT),A
3AAC 3E 00      1048  LD     A,0
3AAE DD 77 00   1049  LD     (IX),A
3AB1            1050 #CHSTEP2
3AB1 DD 6E 01   1051  LD     L,(IX+1)
3AB4 DD 66 02   1052  LD     H,(IX+2)
3AB7 7C         1053  LD     A,H
3AB8 CB 47      1054  BIT    0,A
3ABA 20 04 3E   1055  IF     Z      THEN LD A,"•"  ELSE LD A,"."
3ABD DF 18 02
3AC0 3E A1
3AC2 B7         1056  OR     A
3AC3 CB 1C      1057  RR     H
3AC5 CD 8C 3B   1058  CALL   #PRINT
3AC8            1059 #BASTEP
3AC8 01 07 00   1060  LD     BC,7
3ACB DD 09      1061  ADD    IX,BC
3ACD D9         1062  EXX
3ACE 15         1063  DEC    D
3ACF 7A         1064  LD     A,D
3AD0 D9         1065  EXX
3AD1 FE 00      1066  CP     0
3AD3 C2 C0 39   1067  JP     NZ,#LOOPBA
3AD6 3A 38 40   1068  LD     A,(#HIT)
3AD9 FE 01 C2   1069  IF     A<>1   JP #HISTEP1
3ADC FF 3A
3ADE 3A 2B 40   1070  LD     A,(#PLFM)
3AE1 FE 06 30   1071  IF     A<$6   THEN LD (#PLFMS),A
3AE4 03 32 2C
3AE7 40
3AE8 3E 20      1072  LD     A,32
3AEA 32 2B 40   1073  LD     (#PLFM),A
3AED            1074
3AED 21 26 0C   1075  LD     HL,$0C00+38
3AF0 CD 1E 20   1076  CALL   @LOC
3AF3 3A 2F 40   1077  LD     A,(#PLDM)
3AF6 3C         1078  INC    A
3AF7 32 2F 40   1079  LD     (#PLDM),A
3AFA C6 30      1080  ADD    A,"0"
3AFC CD F4 1F   1081  CALL   @PRINT
3AFF            1082 #HISTEP1
3AFF 3A 38 40   1083  LD     A,(#HIT)
3B02 FE 02 C2   1084  IF     A<>2   JP #HISTEP2
3B05 28 3B
3B07 3A 32 40   1085  LD     A,(#ENFM)
3B0A FE 06 30   1086  IF     A<$6   THEN LD (#ENFMS),A
3B0D 03 32 33
3B10 40
3B11 3E 20      1087  LD     A,32
3B13 32 32 40   1088  LD     (#ENFM),A
3B16            1089
3B16 21 22 0C   1090  LD     HL,$0C00+34
3B19 CD 1E 20   1091  CALL   @LOC
3B1C 3A 36 40   1092  LD     A,(#ENDM)
3B1F 3C         1093  INC    A
3B20 32 36 40   1094  LD     (#ENDM),A
3B23 C6 30      1095  ADD    A,"0"
3B25 CD F4 1F   1096  CALL   @PRINT
3B28            1097 #HISTEP2
3B28 F1         1098  POP    AF
3B29 D1         1099  POP    DE
3B2A C1         1100  POP    BC
3B2B E1         1101  POP    HL
3B2C C9         1102  RET
3B2D 00         1103 #SSS : DB 0
3B2E            1104 #CHECK
3B2E 3A 2F 40   1105  LD     A,(#PLDM)
3B31 FE 08 CA   1106  IF     A=8    JP #ENWIN
3B34 E5 3B
3B36 3A 36 40   1107  LD     A,(#ENDM)
3B39 FE 08 CA   1108  IF     A=8    JP #PLWIN
3B3C EF 3B
3B3E C9         1109  RET
3B3F            1110 #PUT
3B3F            1111  ;   A <---- CHARATER NO.
3B3F            1112  ;   H,L <-- Y,X
3B3F E5         1113  PUSH   HL
3B40 C5         1114  PUSH   BC
3B41 D5         1115  PUSH   DE
3B42 F5         1116  PUSH   AF
3B43 EB         1117  EX     HL,DE
3B44 26 00      1118  LD     H,0
3B46 6F         1119  LD     L,A
3B47 29         1120  ADD    HL,HL
3B48 29         1121  ADD    HL,HL
3B49 01 D4 3C   1122  LD     BC,#CHHEAD
3B4C 09         1123  ADD    HL,BC
3B4D EB         1124  EX     HL,DE
3B4E 1A         1125  LD     A,(DE)
```

▶最近，電気製品から出る高周波が耳につく。X68000のディスプレイ，こたつ，CDプレイヤーなど。これは私の耳の可聴範囲が広がったためであろうか？ 渡辺 一矢(21)石川県

```
3B4F CD 8C 3B  1126  CALL   #PRINT
3B52 2C        1127  INC    L
3B53 13        1128  INC    DE
3B54 1A        1129  LD     A,(DE)
3B55 CD 8C 3B  1130  CALL   #PRINT
3B58 2D        1131  DEC    L
3B59 24        1132  INC    H
3B5A 13        1133  INC    DE
3B5B 1A        1134  LD     A,(DE)
3B5C CD 8C 3B  1135  CALL   #PRINT
3B5F 2C        1136  INC    L
3B60 13        1137  INC    DE
3B61 1A        1138  LD     A,(DE)
3B62 CD 8C 3B  1139  CALL   #PRINT
3B65 F1        1140  POP    AF
3B66 D1        1141  POP    DE
3B67 C1        1142  POP    BC
3B68 E1        1143  POP    HL
3B69 C9        1144  RET
3B6A           1145 #SCRN
3B6A E5        1146  PUSH   HL
3B6B 7C        1147  LD     A,H
3B6C FE 0C 38  1148  IF     A>=12  THEN LD A,32 : POP HL : RET
3B6F 04 3E 20
3B72 E1 C9
3B74 B7 20 04  1149  IF     A=0    THEN LD A,32 : POP HL : RET
3B77 3E 20 E1
3B7A C9
3B7B 7D        1150  LD     A,L
3B7C FE 27 38  1151  IF     A>=39  THEN SUB 38 : LD L,A : LD A,H : ADD A,13 : LD
 H,A
3B7F 07 D6 26
3B82 6F 7C C6
3B85 0D 67
3B87 CD 1B 20  1152  CALL   @SCRN
3B8A E1        1153  POP    HL
3B8B C9        1154  RET
3B8C           1155 #PRINT
3B8C E5        1156  PUSH   HL
3B8D F5        1157  PUSH   AF
3B8E 7C        1158  LD     A,H
3B8F FE 0B 38  1159  IF     A>=11  THEN POP AF : POP HL : RET
3B92 03 F1 E1
3B95 C9
3B96 B7 20 03  1160  IF     A=0    THEN POP AF : POP HL : RET
3B99 F1 E1 C9
3B9C 7D        1161  LD     A,L
3B9D FE 4D 38  1162  IF     A>=77  THEN POP AF : POP HL : RET
3BA0 03 F1 E1
3BA3 C9
3BA4 FE 27 38  1163  IF     A>=39  THEN SUB 38 : LD L,A : LD A,H : ADD A,13 : LD
 H,A
3BA7 07 D6 26
3BAA 6F 7C C6
3BAD 0D 67
3BAF CD 1E 20  1164  CALL   @LOC
3BB2 F1        1165  POP    AF
3BB3 CD F4 1F  1166  CALL   @PRINT
3BB6 E1        1167  POP    HL
3BB7 C9        1168  RET
3BB8           1169 #RND
3BB8           1170  ; RONDOM VER. - A
3BB8           1171  ;
3BB8 C5        1172  PUSH   BC
3BB9 D5        1173  PUSH   DE
3BBA E5        1174  PUSH   HL
3BBB 21 3C 40  1175  LD     HL,#RNDBIT1
3BBE CD DD 3B  1176  CALL   #RND2
3BC1 F5        1177  PUSH   AF
3BC2 D1        1178  POP    DE
3BC3 21 3D 40  1179  LD     HL,#RNDBIT2
3BC6 CD DD 3B  1180  CALL   #RND2
3BC9 F5        1181  PUSH   AF
3BCA C1        1182  POP    BC
3BCB 79        1183  LD     A,C
3BCC AB        1184  XOR    E
3BCD 1F        1185  RRA
3BCE 2A 3E 40  1186  LD     HL,(#RNDBUFF)
3BD1 ED 6A     1187  ADC    HL,HL
3BD3 22 3E 40  1188  LD     (#RNDBUFF),HL
3BD6 E1        1189  POP    HL
3BD7 D1        1190  POP    DE
3BD8 C1        1191  POP    BC
3BD9 3A 3E 40  1192  LD     A,(#RNDBUFF)
3BDC C9        1193  RET
3BDD           1194 #RND2
3BDD 46        1195  LD     B,(HL)
3BDE 2A 3E 40  1196  LD     HL,(#RNDBUFF)
3BE1           1197 #RNDLOOP
3BE1 29        1198  ADD    HL,HL
3BE2 10 FD     1199  DJNZ   #RNDLOOP
3BE4 C9        1200  RET
3BE5           1201 #ENWIN
3BE5 CD 35 3C  1202  CALL   #SLOW
3BE8 D1        1203  POP    DE
3BE9 21 2F 02  1204  LD     HL,$0200+47
3BEC C3 F6 3B  1205  JP     #WINSTEP
3BEF           1206 #PLWIN
3BEF CD 35 3C  1207  CALL   #SLOW
3BF2 D1        1208  POP    DE
3BF3 21 09 02  1209  LD     HL,$0209
3BF6           1210 #WINSTEP
3BF6 11 F8 3E  1211  LD     DE,#WIN
3BF9 01 00 00  1212  LD     BC,0
3BFC           1213 #WINLOOP
3BFC 1A        1214  LD     A,(DE)
3BFD CD 8C 3B  1215  CALL   #PRINT
3C00 13        1216  INC    DE
3C01 2C        1217  INC    L
3C02 04        1218  INC    B
3C03 3E 17     1219  LD     A,23
3C05 B8 20 08  1220  IF     A=B    THEN LD B,0 : INC C : LD A,L : SUB 23 : LD L,
A : INC H
3C08 06 00 0C
3C0B 7D D6 17
3C0E 6F 24
3C10 3E 05     1221  LD     A,5
3C12 B9 C2 FC  1222  IF     A<>C   JP #WINLOOP
3C15 3B
3C16 01 00 FF  1223  LD     BC,$FF00
3C19 CD 9D 3C  1224  CALL   #WAIT
3C1C 21 01 0C  1225  LD     HL,$0C01
3C1F CD 1E 20  1226  CALL   @LOC
3C22 11 4D 3D  1227  LD     DE,#ANYKEY
3C25 CD E8 1F  1228  CALL   @MSG
3C28 CD C4 1F  1229  CALL   @BELL
3C2B           1230 #WINLOOP2
3C2B CD D0 1F  1231  CALL   @GETKY
3C2E B7 CA 2B  1232  IF     A=0    JP #WINLOOP2
3C31 3C
3C32 C3 07 30  1233  JP     #HOTSTART
3C35           1234 #SLOW
3C35 06 19     1235  LD     B,25
3C37           1236 #SLLOOP
3C37 C5        1237  PUSH   BC
3C38 CD 3D 30  1238  CALL   #PLAYER
3C3B CD 65 33  1239  CALL   #ENEMY
3C3E CD AF 39  1240  CALL   #BALL
3C41 01 00 40  1241  LD     BC,$4000
3C44 CD 9D 3C  1242  CALL   #WAIT
3C47 C1        1243  POP    BC
3C48 10 ED     1244  DJNZ   #SLLOOP
3C4A C9        1245  RET
3C4B           1246 #INIT
3C4B 3E 28     1247  LD     A,40
3C4D CD 30 20  1248  CALL   @WIDCH
3C50 C9        1249  RET
3C51           1250 #BALLINIT
3C51 21 8F 40  1251  LD     HL,#MUDBALL
3C54 06 70     1252  LD     B,16*7
3C56 3E 00     1253  LD     A,0
3C58           1254 #BILOOP
3C58 77        1255  LD     (HL),A
3C59 23        1256  INC    HL
3C5A 10 FC     1257  DJNZ   #BILOOP
3C5C C9        1258  RET
3C5D           1259 #SCREEN
3C5D 3E 0C     1260  LD      A,$0C
3C5F CD F4 1F  1261  CALL   @PRINT
3C62 06 18     1262  LD     B,24
3C64 21 27 00  1263  LD     HL,39
3C67           1264 #LOOPIN
3C67 CD 1E 20  1265  CALL   @LOC
3C6A 3E 7B     1266  LD     A,"{"
3C6C CD F4 1F  1267  CALL   @PRINT
3C6F CD F4 1F  1268  CALL   @PRINT
3C72 24        1269  INC    H
3C73 10 F2     1270  DJNZ   #LOOPIN
3C75 11 AB 3C  1271  LD     DE,#BAR
3C78 21 00 00  1272  LD     HL,0
3C7B CD 1E 20  1273  CALL   @LOC
3C7E CD E8 1F  1274  CALL   @MSG
3C81 21 00 0B  1275  LD     HL,$0B00
3C84 CD 1E 20  1276  CALL   @LOC
3C87 CD E8 1F  1277  CALL   @MSG
3C8A 21 00 0D  1278  LD     HL,$0D00
3C8D CD 1E 20  1279  CALL   @LOC
3C90 CD E8 1F  1280  CALL   @MSG
3C93 21 00 18  1281  LD     HL,$1800
3C96 CD 1E 20  1282  CALL   @LOC
3C99 CD E8 1F  1283  CALL   @MSG
3C9C C9        1284  RET
3C9D           1285 #WAIT
3C9D C5        1286  PUSH   BC
3C9E           1287 #LOOPW1
3C9E 00        1288  NOP
3C9F 00        1289  NOP
3CA0 00        1290  NOP
3CA1 00        1291  NOP
3CA2 3E 00     1292  LD     A,0
3CA4 0B        1293  DEC    BC
3CA5 B8        1294  CP     B
3CA6 C2 9E 3C  1295  JP     NZ,#LOOPW1
3CA9 C1        1296  POP    BC
3CAA C9        1297  RET
3CAB           1298 #BAR
3CAB 7B 7B 7B  1299  DM     "{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{"
3CAE 7B 7B 7B
3CB1 7B 7B 7B
3CB4 7B 7B 7B
3CB7 7B 7B 7B
3CBA 7B 7B 7B
3CBD 7B 7B 7B
3CC0 7B 7B 7B
3CC3 7B 7B 7B
3CC6 7B 7B 7B
3CC9 7B 7B 7B
3CCC 7B 7B 7B
3CCF 7B 7B 7B
3CD2 0D        1300  DB     $0D
3CD3 0D        1301  DB  $0D
3CD4           1302 #CHHEAD
3CD4 20 20 20  1303  DB  $20 : $20 : $20 : $20
3CD7 20
3CD8 3C A8 2D  1304  DB  $3C : $A8 : $2D : $29
3CDB 29
3CDC 20 28 20  1305  DB  $20 : $28 : $20 : $C7
3CDF C7
3CE0 28 DA 2F  1306  DB  $28 : $DA : $2F : $49
3CE3 49
3CE4 2F A8 29  1307  DB  $2F : $A8 : $29 : $29
3CE7 29
3CE8 2F A8 2F  1308  DB  $2F : $A8 : $2F : $28
3CEB 28
3CEC 2F A8 C9  1309  DB  $2F : $A8 : $C9 : $49
3CEF 49
3CF0 2D A8 C9  1310  DB  $2D : $A8 : $C9 : $49
3CF3 49
3CF4 20 2F 2F  1311  DB  $20 : $2F : $2F : $29
3CF7 29
3CF8 20 2D 2F  1312  DB  $20 : $2D : $2F : $29
3CFB 29
3CFC 2D 3E 28  1313  DB  $2D : $3E : $28 : $CD
3CFF CD
3D00 29 20 C4  1314  DB  $29 : $20 : $C4 : $20
3D03 20
3D04 2F 29 49  1315  DB  $2F : $29 : $49 : $2D
3D07 2D
3D08 2F A8 28  1316  DB  $2F : $A8 : $28 : $28
3D0B 28
3D0C 2F A8 2F  1317  DB  $2F : $A8 : $2F : $29
3D0F 29
3D10 2C CD 49  1318  DB  $2C : $CD : $49 : $29
3D13 29
3D14 2C 2D 28  1319  DB  $2C : $2D : $28 : $28
3D17 28
3D18 CD 20 2F  1320  DB  $CD : $20 : $2F : $28
3D1B 28
3D1C 2D 20 2F  1321  DB  $2D : $20 : $2F : $28
3D1F 28
3D20 CD 20 20  1322  DB  $CD : $20 : $20 : $29
3D23 29
3D24 2D 3C 20  1323  DB  $2D : $3C : $20 : $20
3D27 20
3D28 20 20 2D  1324  DB  $20 : $20 : $2D : $3D
3D2B 3D
3D2C 20 C9 28  1325  DB  $20 : $C9 : $28 : $20
3D2F 20
3D30 3E 2D 20  1326  DB  $3E : $2D : $20 : $20
3D33 20
3D34 20 20 3D  1327  DB  $20 : $20 : $3D : $2D
3D37 2D
3D38 20 20 2F  1328  DB  $20 : $20 : $2F : $68
3D3B 68
3D3C           1329 #PLFMTAB
3D3C 01 02 03  1330  DB  $1 : $2 : $3 : $2
3D3F 02
3D40 04 05     1331  DB  $4 : $5
3D42           1332 #ENFMTAB
3D42 0B 0A 0B  1333  DB  $B : $A : $B : $C
3D45 0C
3D46 0D 0E     1334  DB  $D : $E
3D48 DC BE 96  1335 #DIFICULTY : DB 220 : 190 : 150 : 100 : 40
3D4B 64 28
3D4D           1336 #ANYKEY
3D4D 20 20 20  1337  DM     "               HIT ANY KEY
3D50 20 20 20
3D53 20 20 20
3D56 20 20 20
3D59 20 48 49
3D5C 54 20 41
3D5F 4E 59 20
3D62 4B 45 59
3D65 20 20 20
3D68 20 20 20
3D6B 20 20 20
3D6E 20 20 20
3D71 20 20
3D73 0D        1338  DB     $0D
3D74           1339 #TITLE
3D74           1340 ;      "                                   ------------------
------------------
3D74           1341 ;    "                        -------------------------
----------
3D74 D6 D6 20  1342  DM "ﾖﾖ   ﾖﾖ ﾖﾖ   ﾖﾖ ﾖﾖﾖﾖﾖﾖ        ﾖﾖﾖﾖﾖ   ﾖﾖﾖﾖ  ﾖﾖ    ﾖﾖ
 ﾖﾖ ﾖﾖ   ﾖﾖ ﾖﾖ"
3D77 20 20 D6
3D7A D6 20 D6
3D7D D6 20 20
3D80 20 D6 D6
```

▶シムシティーの長方形の車たちは，どっから現れてどこへいくんでしょう。見ていると梅田の動く歩道のようだ。　　後藤　正和(19)大阪府

```
3D83 20 D6 D6
3D86 D6 D6 D6
3D89 D6 20 20
3D8C 20 20 20
3D8F 20 20 20
3D92 D6 D6 D6
3D95 D6 D6 20
3D98 20 20 D6
3D9B D6 D6 D6
3D9E 20 20 D6
3DA1 D6 20 20
3DA4 20 20 D6
3DA7 D6 20 20
3DAA 20 20 D6
3DAD D6 20 D6
3DB0 D6 20 20
3DB3 D6 D6 20
3DB6 D6 D6
3DB8 D6 D6 D6  1343  DM "ﾖﾖﾖ ﾖﾖﾖ ﾖﾖ  ﾖﾖ ﾖﾖ   ﾖﾖ       ﾖﾖ  ﾖﾖ ﾖﾖ  ﾖﾖ ﾖﾖ    ﾖﾖ
 ﾖﾖ ﾖﾖﾖ ﾖﾖ  ﾖ"
3DBB 20 D6 D6
3DBE D6 20 D6
3DC1 D6 20 20
3DC4 20 D6 D6
3DC7 20 D6 D6
3DCA 20 20 20
3DCD D6 D6 20
3DD0 20 20 20
3DD3 20 20 20
3DD6 D6 D6 20
3DD9 20 D6 D6
3DDC 20 D6 D6
3DDF 20 20 D6
3DE2 D6 20 D6
3DE5 D6 20 20
3DE8 20 20 D6
3DEB D6 20 20
3DEE 20 20 D6
3DF1 D6 20 D6
3DF4 D6 D6 20
3DF7 D6 D6 20
3DFA 20 D6
3DFC D6 D6 20  1344  DM "ﾖﾖ ﾖ ﾖﾖ ﾖﾖ  ﾖﾖ ﾖﾖ   ﾖﾖ       ﾖﾖﾖﾖﾖ  ﾖﾖ  ﾖﾖ ﾖﾖ    ﾖﾖ
 ﾖﾖ ﾖﾖﾖﾖﾖﾖ ﾖ "
3DFF D6 20 D6
3E02 D6 20 D6
3E05 D6 20 20
3E08 20 D6 D6
3E0B 20 D6 D6
3E0E 20 20 20
3E11 D6 D6 20
3E14 20 20 20
3E17 20 20 20
3E1A D6 D6 D6
3E1D D6 D6 20
3E20 20 D6 D6
3E23 20 20 D6
3E26 D6 20 D6
3E29 D6 20 20
3E2C 20 20 D6
3E2F D6 20 20
3E32 20 20 D6
3E35 D6 20 D6
3E38 D6 D6 D6
3E3B D6 D6 20
3E3E D6 20
3E40 D6 D6 20  1345  DM "ﾖﾖ   ﾖﾖ ﾖﾖ   ﾖﾖ ﾖﾖ   ﾖﾖ       ﾖﾖ  ﾖﾖ ﾖﾖﾖﾖﾖﾖ ﾖﾖ    ﾖﾖ
 ﾖﾖ ﾖﾖ ﾖﾖﾖ     "
3E43 20 20 D6
3E46 D6 20 D6
3E49 D6 20 20
3E4C 20 D6 D6
3E4F 20 D6 D6
3E52 20 20 20
3E55 D6 D6 20
3E58 20 20 20
3E5B 20 20 20
3E5E D6 D6 20
3E61 20 D6 D6
3E64 20 D6 D6
3E67 D6 D6 D6
3E6A D6 20 D6
3E6D D6 20 20
3E70 20 20 D6
3E73 D6 20 20
3E76 20 20 D6
3E79 D6 20 D6
3E7C D6 20 D6
3E7F D6 D6 20
3E82 20 20
3E84 D6 D6 20  1346  DM "ﾖﾖ   ﾖﾖ  ﾖﾖﾖﾖﾖ  ﾖﾖﾖﾖﾖﾖ       ﾖﾖﾖﾖﾖ  ﾖﾖ  ﾖﾖ ﾖﾖﾖﾖﾖ ﾖﾖﾖﾖﾖ
 ﾖﾖ ﾖﾖ  ﾖﾖ    "
3E87 20 20 D6
3E8A D6 20 20
3E8D D6 D6 D6
3E90 D6 D6 20
3E93 20 D6 D6
3E96 D6 D6 D6
3E99 D6 20 20
3E9C 20 20 20
3E9F 20 20 20
3EA2 D6 D6 D6
3EA5 D6 D6 20
3EA8 20 D6 D6
3EAB 20 20 D6
3EAE D6 20 D6
3EB1 D6 D6 D6
3EB4 D6 20 D6
3EB7 D6 D6 D6
3EBA D6 20 D6
3EBD D6 20 D6
3EC0 D6 20 20
3EC3 D6 D6 20
3EC6 20 20
3EC8 D6 D6 D6  1347 #DEMOMUD : DM "ﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖ" : DB $0D
3ECB D6 D6 D6
3ECE D6 D6 D6
3ED1 D6 D6 D6
3ED4 D6 D6 D6
3ED7 D6 D6 D6
3EDA D6 D6 D6
3EDD D6 D6 0D
3EE0 AD D6 D6  1348 #DEMOMUD': DM "ｭﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖﾖｭ" : DB $0D
3EE3 D6 D6 D6
3EE6 D6 D6 D6
3EE9 D6 D6 D6
3EEC D6 D6 D6
3EEF D6 D6 D6
3EF2 D6 D6 D6
3EF5 D6 AD 0D
3EF8           1349 #WIN
3EF8 D6 D6 20  1350  DM "ﾖﾖ   ﾖﾖ ﾖﾖ ﾖﾖ   ﾖﾖ   ﾖﾖ"
3EFB 20 20 D6
3EFE D6 20 D6
3F01 D6 20 D6
3F04 D6 20 20
3F07 20 D6 D6
3F0A 20 20 20
3F0D D6 D6
3F0F D6 D6 20  1351  DM "ﾖﾖ ﾖ ﾖﾖ ﾖﾖ ﾖﾖﾖ  ﾖﾖ  ﾖﾖﾖ"
3F12 D6 20 D6
3F15 D6 20 D6
3F18 D6 20 D6
3F1B D6 D6 20
3F1E 20 D6 D6
3F21 20 20 D6
3F24 D6 D6
3F26 D6 D6 20  1352  DM "ﾖﾖ ﾖ ﾖﾖ ﾖﾖ ﾖﾖﾖﾖﾖﾖﾖ  ﾖﾖ "
3F29 D6 20 D6
3F2C D6 20 D6
3F2F D6 20 D6
3F32 D6 D6 D6
3F35 D6 D6 D6
3F38 20 20 D6
3F3B D6 20
3F3D D6 D6 D6  1353  DM "ﾖﾖﾖﾖﾖﾖﾖ ﾖﾖ ﾖﾖ  ﾖﾖﾖ      "
3F40 D6 D6 D6
3F43 D6 20 D6
3F46 D6 20 D6
3F49 D6 20 20
3F4C D6 D6 D6
3F4F 20 20 20
3F52 20 20
3F54 20 D6 D6  1354  DM " ﾖﾖ ﾖﾖ  ﾖﾖ ﾖﾖ   ﾖﾖ ﾖﾖ  "
3F57 20 D6 D6
3F5A 20 20 D6
3F5D D6 20 D6
3F60 D6 20 20
3F63 20 D6 D6
3F66 20 D6 D6
3F69 20 20
3F6B           1355 #GRFORM
3F6B 00 00 00  1356  DB $00:$00:$00:$00:$00:$00:$00:$00:$00:$00
3F6E 00 00 00
3F71 00 00 00
3F74 00
3F75 00 00 00  1357  DB $00:$00:$00:$00:$00:$00:$00:$00:$00
3F78 00 00 00
3F7B 00 00 00
3F7E 00 00 00  1358  DB $00:$00:$00:$12:$34:$56:$78:$9A:$BC:$DE
3F81 12 34 56
3F84 78 9A BC
3F87 DE
3F88 FF ED CB  1359  DB $FF:$ED:$CB:$A9:$87:$65:$43:$21:$00
3F8B A9 87 65
3F8E 43 21 00
3F91 00 00 12  1360  DB $00:$00:$12:$23:$45:$67:$89:$AB:$CD:$EF
3F94 23 45 67
3F97 89 AB CD
3F9A EF
3F9B FF 00 00  1361  DB $FF:$00:$00:$00:$00:$00:$00:$00:$00
3F9E 00 00 00
3FA1 00 00 00
3FA4 32 10 01  1362  DB $32:$10:$01:$23:$FF:$EE:$DC:$BA:$98:$76
3FA7 23 FF EE
3FAA DC BA 98
3FAD 76
3FAE 54 32 10  1363  DB $54:$32:$10:$00:$00:$00:$00:$00:$00
3FB1 00 00 00
3FB4 00 00 00
3FB7 00 01 22  1364  DB $00:$01:$22:$2E:$EE:$EE:$EE:$EE:$00:$00
3FBA 2E EE EE
3FBD EE EE 00
3FC0 00
3FC1 00 00 00  1365  DB $00:$00:$00: )0:$00:$00:$00:$00:$00
3FC4 00 00 00
3FC7 00 00 00
3FCA 00 01 23  1366  DB $00:$01:$23:s45:$67:$89:$AB:$00:$44:$44
3FCD 45 67 89
3FD0 AB 00 44
3FD3 44
3FD4 55 43 21  1367  DB $55:$43:$21:$10:$00:$00:$00:$00:$00
3FD7 10 00 00
3FDA 00 00 00
3FDD 66 66 66  1368  DB $66:$66:$66:  6:$66:$66:$66:$66:$66:$66
3FE0 66 66 66
3FE3 66 66 66
3FE6 66
3FE7 66 66 00  1369  DB $66:$66:$00:$00:$00:$00:$00:$00:$00
3FEA 00 00 00
3FED 00 00 00
3FF0 88 66 88  1370  DB $88:$66:$88:$66:$88:$66:$88:$66:$88:$66
3FF3 66 88 66
3FF6 88 66 88
3FF9 66
3FFA 88 66 88  1371  DB $88:$66:$88:$53:$10:$00:$00:$00:$00
3FFD 53 10 00
4000 00 00 00
4003 FF DB 97  1372  DB $FF:$DB:$97:$53:$10:$00:$EE:$EE:$EE:$EE
4006 53 10 00
4009 EE EE EE
400C EE
400D EE EE EE  1373  DB $EE:$EE:$EE:$00:$00:$00:$00:$00:$00
4010 00 00 00
4013 00 00 00
4016 FF DB 97  1374  DB $FF:$DB:$97:$53:$31:$00:$01:$33:$57:$9B
4019 53 31 00
401C 01 33 57
401F 9B
4020 DF F0 00  1375  DB $DF:$F0:$00:$00:$00:$00:$00:$00:$00
4023 00 00 00
4026 00 00 00
4029 03        1376 #PLX    : DB 3
402A 09        1377 #PLY    : DB 9
402B 02        1378 #PLFM   : DB 2
402C 00        1379 #PLFMS  : DB 0
402D 00        1380 #STOCK  : DB 0
402E 00        1381 #PLAIM  : DB 0
402F 00        1382 #PLDM   : DB 0
4030 97        1383 #ENX    : DB 151
4031 09        1384 #ENY    : DB 9
4032 02        1385 #ENFM   : DB 2
4033 00        1386 #ENFMS  : DB 0
4034 00        1387 #STOCKE : DB 0
4035 00        1388 #ENMOVE : DB 0
4036 00        1389 #ENDM   : DB 0
4037 00        1390 #LASTKY : DB 0
4038 00        1391 #HIT    : DB 0
4039 00        1392 #CUX    : DB 0
403A 00        1393 #CHCUR  : DB 0
403B 00        1394 #MOVE   : DB 0
403C           1395 #RNDBIT1
403C 10        1396  DB 16
403D           1397 #RNDBIT2
403D 02        1398  DB 2
403E           1399 #RNDBUFF
403E 14 56     1400  DW $5614
4040           1401 #SIDE
4040 00        1402  DB    0
4041           1403 #CONST
4041 00 00 00  1404  DS 78
4044 00 00 00
4047 00 00 00
404A 00 00 00
404D 00 00 00
4050 00 00 00
4053 00 00 00
4056 00 00 00
4059 00 00 00
405C 00 00 00
405F 00 00 00
4062 00 00 00
4065 00 00 00
4068 00 00 00
406B 00 00 00
406E 00 00 00
4071 00 00 00
4074 00 00 00
4077 00 00 00
407A 00 00 00
407D 00 00 00
4080 00 00 00
4083 00 00 00
4086 00 00 00
4089 00 00 00
408C 00 00 00
408F           1405 #MUDBALL
408F 00 00 00  1406  DS 7*16
4092 00 00 00
4095 00 00 00
```


▶私はついにコンピュータクラブの部長になることになってしまいました。プログラムもろくに打てないのに……。でも，私がいちばんえらいんだ。みんな私の前でひざまずくことになるであろう。
市川　徳明(17)東京都

```
4098 00 00 00
409B 00 00 00
409E 00 00 00
40A1 00 00 00
40A4 00 00 00
40A7 00 00 00
40AA 00 00 00
40AD 00 00 00
40B0 00 00 00
40B3 00 00 00
40B6 00 00 00
40B9 00 00 00
40BC 00 00 00
40BF 00 00 00
40C2 00 00 00
40C5 00 00 00
40C8 00 00 00
40CB 00 00 00
40CE 00 00 00
40D1 00 00 00
40D4 00 00 00
40D7 00 00 00
40DA 00 00 00
40DD 00 00 00
40E0 00 00 00
40E3 00 00 00
40E6 00 00 00
40E9 00 00 00
40EC 00 00 00
40EF 00 00 00
40F2 00 00 00
40F5 00 00 00
40F8 00 00 00
40FB 00 00 00
40FE 00
40FF 00 00        1407 SPBUF:  DS    2
```

全機種共通システムインデックス

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

X68000CARDDRV用カードゲーム

EIGHT

Iketani Masahiko 池谷 昌彦

「EIGHT」といってもあまり有名なゲームではありませんが，ルール自体は「PAGE ONE」などのメジャーなゲームに似ていますので初めての人でもすぐにルールを覚えてしまえるでしょう。もちろんコンパイルにも対応しています。

実行方法

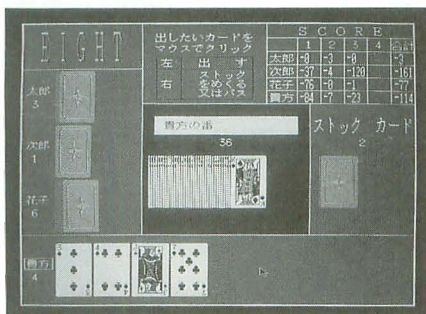
このゲームの実行には1991年1月号の付録ディスクに収録されていたCARDDRV XとCARD2.FNCが必要です。従来のCARD.FNCでもまったく支障なく実行できますが，できるだけ新しいシステム（CARDDRVとCARD2.FNC）を使うようにしてください。

まず，BASICを立ち上げてリスト1を打ち込んでください。

A>CARDDRV TR.DAT

のようにCARDDRVでTR.DATを組み込み，CARD2.FNCを組み込んだBASICから（CARD2.FNCをBASICがあるディレクトリに置き，BASIC.CNFに“FUNC=CARD2”の1行を加える）実行してください。これでゲームが始まります。

この時点でメモリの足りなくなった人はRAMDISKや不要なデバイスドライバをはずしてシステムを再起動してください。このゲームの実行に必要なものは数値演算ドライバとOPMドライバだけです。

コンパイルするときにはリスト中の注釈に従って変更を加えてください。CARDDRV用のライブラリを組み込むのを忘れないように。

ルール解説

では，最初にルールの説明をしておきましょう。プレイヤーは台札と同じスート（カードの種類）か同じ数字のものを出していきます。5枚ずつの手札から始めて手札を早くなくした人がその場は勝ちます。手札のなかに出せるものがないときは出せるものが現れるまで場札を引き続けなければいけません（場札がなくなればパスになります）。こうして1ゲーム終わった時点で残り札から得点を計算し次のゲームに移っていくのです。

ゲームの名前からもわかるように，このゲームでは8のカードは特殊な意味を持ちます。これはいつでも出すことができ，それ以降のカードのスートを指定することができるのです。ただし，このカードは1ゲーム終了時の得点計算で大きな負担となりますので，いつまでも温存していることは必ずしも得策とはいえません。

ゲームを始める前に簡単なルール説明も読めますので，とりあえずゲームを始めてみるのがいいでしょう。

プログラムについて

今回のプログラムでは各人の持てるカードの最大数を26枚とし，これを超えるようなときにはノーカウントとして再ゲームとしましたが，私のやった範囲では実際にそれだけのカードが溜まったことはありません。また，誰も出せるカードがなく，台札もなくなったときにも，やはり再ゲームとしました。

少々長めのプログラムですので入力，解析はやりにくいかもしれません。ぜひ長さにめげずチャレンジしてください。

リスト1

```
 10 /*
 20 /* EIGHT
 30 /*   Programmed by M.I.,Sep.11,'90:revised Jan.12,'91
 40 /*
 50 screen 1,1,1,1:console ,,0
 60 int jj,p
 70 int f=0,g=0,rd=1,pas=0,pa=0
 80 dim cc(51),c(3,25),s(31),m(5)
 90 dim int ten(4,3)
100 dim str nam(3)={ "太郎","次郎","花子","貴方" }
110 dim str suit(3)={ "スペード","ハート  ","ダイヤ  ","クラブ
" }
120 palet(1,0)
130 /* main program
140 while f<>1
150    scrn()
160    play()
170    judge()
180 endwhile
190 owari()
200 end
210 /*
220 /* preparation of screen
230 /*
240 func scrn()
250    int i,j
260    for i=0 to 4:for j=0 to 3:ten(i,j)=0:next:next
270    apage(3):vpage(15)
280    fill(0,0,511,511,10)
290    apage(2)
300    box(0,0,511,511,15):box(1,1,510,510,15)
310    line(2,384,509,384,15):line(160,2,160,383,15)
320    line(2,80,159,80,15):line(161,144,509,144,15)
330    line(320,2,320,143,15):line(376,145,376,383,15)
340    for i=0 to 4:line(321,i*24+24,509,i*24+24,15):next
350    for i=0 to 4:line(i*30+360,25,i*30+360,143,15):next
360    symbol(26,11,"E  I  G  H  T",1,4,1,1,0):symbol(24,10,"
E  I  G  H  T",1,4,1,13,0)
370    symbol(360,6,"S  C  O  R  E",1,1,1,15,0)
380    for i=0 to 3:symbol(324,i*24+53,nam(i),1,1,1,15,0):next
390    for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
400    symbol(481,29,"合計",1,1,1,15,0)
410 endfunc
420 /*
430 /* play
440 /*
450 func play()
460    for rd=1 to 4
```

```
470       prep()
480       splay()
490       jd()
500    next
510 endfunc
520 /*
530 /* preparation
540 /*
550 func prep()
560    int i,j,a,b,k
570 /* music data set
580    if rd=1 then {
590       m_init()
600       for i=1 to 8:m_alloc(i,2000):m_assign(i,i):next
610       m_trk(1,"q2@23v12o2t200e64")
620       m_trk(2,"q2@23v12o2t200c64")
630       m_trk(3,"q6@56v15o5t100l16aeae")
640       m_trk(4,"q8@43v10o3t100a4")
650       m_trk(5,"q8@57v14o6t100c32r64>a4")
660       m_trk(6,"q4@15v9 o3t200c16v7c8")
670       m_trk(7,"q8@30v10o3t120l8fc")
680       m_trk(8,"q8@52v15o4t80 c2")
690    }
700 /* deal
710    apage(0):fill(0,0,511,511,0)
720    apage(1):fill(0,0,511,511,0)
730    fill((rd-1)*30+331,25,(rd-1)*30+359,47,0)
740    fill(rd*30+331,25,rd*30+359,47,5)
750    for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
760    if rd=1 then {
770       symbol(184,40,"ルールの説明は",1,1,1,15,0)
780       p=sel(176,96,1,1):if p=1 then rule()
790    }
800    randomize(val(mid$(time$,4,2)+right$(time$,2)))
810    for i=0 to 51:cc(i)=i+1:next
820    for i=0 to 3:m(i)=5:next
830    m(4)=32:m(5)=0
840    if p=2 or rd>1 then { /*C に変換する時は 840-890 削除
850       erupms()
860       symbol(200,24,"シャッフル",1,1,1,15,0)
870       symbol(224,56,"及び",1,1,1,15,0)
880       symbol(200,88,"カード配布",1,1,1,15,0)
890    }
900    for i=0 to 99
910       a=int(rnd()*52):b=int(rnd()*52)
920       k=cc(a):cc(a)=cc(b):cc(b)=k
930    next
940    for i=0 to 4
950       c(0,i)=cc(i)
960       c(1,i)=cc(i+5)
970       c(2,i)=cc(i+10)
980       c(3,i)=cc(i+15)
990    next
1000    for i=0 to 31:s(i)=cc(i+20):next
1010    for i=0 to 51:cc(i)=0:next
1020    for i=0 to 3
1030       for j=5 to 25
1040          c(i,j)=0
1050       next
1060    next
1070    erupms()
1080    if rd=1 and p=1 then {
1090       click()
1100       apage(0):fill(0,0,511,511,0):apage(1)
1110    }
1120    mkba():symbol(8,428,nam(3),1,1,1,15,0)
1130    plcd()
1140    for i=0 to 2:comcd(i):next
1150    stcd()
1160 /* stock card to ba
1170    symbol(176,40,"ストックカードを",1,1,1,15,0)
1180    symbol(200,72,"めくります",1,1,1,15,0)
1190    m(4)=m(4)-1   /* ストックカードの枚数
1200    m(5)=m(5)+1   /* 場のカードの枚数
1210    min_st(m(4))
1220    bacd(s(m(4)),m(5))
1230    cc(m(5)-1)=s(m(4))
1240    wait(40)
1250    erupms()
1260 /* play order
1270    if rd=1 then {
1280       symbol(184,24,"順番を決めます",1,1,1,15,0)
1290       symbol(176,56,"いい時にマウスを",1,1,1,15,0)
1300       symbol(208,88,"クリック",1,1,1,15,0)
1310       mouse(1)
1320       symbol(264,177,"が最初",1,1,1,1,0)
1330       repeat
1340          jj=int(rnd()*4)
1350          fill(224,177,256,192,15)
1360          symbol(224,177,nam(jj),1,1,1,1,0)
1370          msstat(i,j,a,b)
1380       until a=-1 or b=-1
1390       mouse(0)
1400       wait(50):erupms():fill(224,176,320,192,0)
1410    } else {
1420       symbol(224,177,nam(jj)+" が最初",1,1,1,1,0)
1430       wait(60):fill(224,176,320,192,0) }
1440 endfunc
1450 /*
1460 /* subplay
1470 /*
1480 func splay()
1490    repeat
1500       switch jj
1510          case 0 : com(0):if g=1 then break
1520          case 1 : com(1):if g=1 then break
1530          case 2 : com(2):if g=1 then break
1540          case 3 : you()
1550       endswitch
1560    until g=1
1570 endfunc
1580 /*
1590 /* judge1
1600 /*
1610 func jd()
1620    int i,p
1630    m_play(8)
1640    apage(0):vpage(14)
1650    fill(2,81,159,143,10):fill(2,385,509,509,10)
1660    box(0,144,511,385,15):box(1,145,510,384,15):fill(2,146,5
09,383,12)
1670    line(2,176,509,176,15):for i=0 to 3:line(2,i*40+224,509,
i*40+224,15):next
1680    fill(2,177,509,223,6)
1690    for i=0 to 10:line(i*32+96,176,i*32+96,384,15):next
1700    line(456,176,456,384,15):line(48,176,48,384,15,&HAAAA)
1710    line(48,200,456,200,15,&HAAAA)
1720    for i=0 to 3:line(48,i*40+244,509,i*40+244,15,&HAAAA):ne
xt
1730    symbol(128,152,"第"+str$(rd)+"回  計算結果",2,1,1,15,0)
1740    symbol(2,192,"カード",1,1,1,15,0):symbol(56,180,"種類",1
,1,1,15,0)
1750    symbol(64,204,"点",1,1,1,15,0):symbol(462,192,"合  計",1
,1,1,15,0)
1760    for i=0 to 3:symbol(8,i*40+234,nam(i),1,1,1,15,0)
1770             symbol(48,i*40+226,"手持数",1,1,1,15,0)
1780             symbol(56,i*40+246,"失点",1,1,1,15,0)
1790    next
1800    symbol(108,180,"A   2   3   4   5   6   7   8   9   10
J-K",1,1,1,15,0)
1810    symbol(101,204,"-1  -2  -3  -4  -5  -6  -7  -50 -9  -10
-10",1,1,1,15,0)
1820    for i=0 to 3:cal(i):next
1830    for i=0 to 3:symbol(472,i*40+246,"-"+str$(ten(rd,i)),1,1
,1,15,0):next
1840    apage(2)
1850    for i=0 to 3:symbol(rd*30+334,i*24+48,"-"+str$(ten(rd,i)
),1,2,0,15,0):next
1860    for i=0 to 3
1870       ten(0,i)=ten(1,i)+ten(2,i)+ten(3,i)+ten(4,i)
1880    next
1890    for i=0 to 3
1900       fill(481,i*24+49,509,i*24+71,0)
1910       symbol(485,i*24+48,"-"+str$(ten(0,i)),1,2,0,15,0)
1920    next
1930    apage(1):fill(0,0,511,511,0)
1940    apage(0)
1950    vpage(15)
1960    for i=0 to 3:if ten(rd,i)=0 then jj=i:break
1970    next
1980    if rd<4 then {
1990       symbol(176,40,str$(rd+1)+" 回目に移ります",1,1,1,15,0
)
2000       p=sel(176,96,2,2):if p=1 then g=0 else g=1:f=1:rd=5
2010    } else wait(80)
2020 endfunc
2030 /*
2040 /* judge2
2050 /*
2060 func judge()
2070    int i,j,k,s,top
2080    str a$,b$,c$,d$,e$
2090    dim int te(3)
2100    ten(0,3)=ten(0,3)-1
2110    for i=0 to 3:te(i)=ten(0,i):next
2120    for i=0 to 2:for j=i+1 to 3
2130       if te(i)>te(j) then k=te(i):te(i)=te(j):te(j)=k
2140    next:next
2150    for i=0 to 3:if te(0)=ten(0,i) then top=i:break
2160    next
2170    ten(0,3)=ten(0,3)+1
2180    apage(2)
2190    fill(481,top*24+49,509,top*24+71,5)
2200    symbol(485,top*24+48,"-"+str$(ten(0,top)),1,2,0,15,0)
2210    apage(1)
2220    fill(0,0,511,511,0)
2230    apage(0)
2240    box(0,144,511,385,15):box(1,145,510,384,15):fill(2,146,5
09,383,4)
2250    for i=0 to 5
2260       box(16+i*10,160+i*7,496-i*10,368-i*7,15)
2270    next
2280    symbol(83,228,nam(top)+"の勝ち",3,3,2,5,0)
2290    a$="l4gb2bf#2f#8.a16aae2.f#a2ae2e8.g16ggd2."
2300    b$="l4r4bbr4bbr4aar4aar4<ccr4cc>r4bbr4bb"
2310    c$="l4r4ggr4ggr4eer4eer4aar4aar4ggr4gg"
2320    d$="l4r4ddr4ddr4ccr4ccr4f#f#r4f#f#r4ddr4dd"
2330    e$="l4dr2dr2>ar2ar2<dr2dr2dr2dr2"
2340    m_trk(4,"q7@21v13t170o5"+a$)
2350    m_trk(5,"q3@35v12t170o4"+b$)
2360    m_trk(6,"q3@35v12t170o4"+c$)
2370    m_trk(7,"q3@35v12t170o4"+d$)
2380    m_trk(8,"q2@37v13t170o3"+e$)
2390    m_play(4,5,6,7,8)
2400    if f=0 then {
2410       symbol(352,440,"もう一度やりますか",1,1,1,15,0)
2420       s=sel(380,465,2,2)
2430       if s=1 then {
2440          rd=1:g=0:fill(0,0,511,511,0)
2450          apage(1):fill(0,0,511,511,0)
2460          apage(2):fill(0,0,511,511,0)
2470          vpage(15)
2480       } else f=1
2490    } else wait(100)
2500 endfunc
2510 /* owari()
2520 func owari()
2530    vpage(2):apage(1)
2540    fill(0,0,511,511,2)
2550    symbol(272,400,"お疲れ様でした",1,1,2,15,0)
2560 endfunc
2570 /*
2580 func cal(cj)
2590    int i,ii,mm=0
2600    dim int ma(12),te(12)
2610    for i=0 to 12:ma(i)=0:next
2620    for i=0 to 9:te(i)=i+1:next
2630    te(7)=50
2640    for i=10 to 12:te(i)=10:next
2650    for i=0 to m(cj)-1
2660       for ii=0 to 12
2670          if (c(cj,i)-1) mod 13 = ii then {
2680             ma(ii)=ma(ii)+1:break
2690          }
2700       next
2710    next
```

▶ついに，X68000用のハンディスキャナが出た。これでマウスで四苦八苦して描く苦労ともおさらばだ。よおし，レッツゴー秋葉原だ。大学入ってから（苦笑）。

富田　祐樹(18)東京都

```
2720     for i=0 to 12
2730        te(i)=ma(i)*te(i)
2740        ten(rd,cj)=ten(rd,cj)+te(i)
2750     next
2760     for i=0 to 9
2770        if ma(i)>0 then {
2780           symbol(i*32+108,cj*40+226,str$(ma(i)),1,1,1,15,0)
2790           symbol(i*32+104,cj*40+246,"-"+str$(te(i)),1,1,1,15
,0)
2800        }
2810     next
2820     if ma(10)+ma(11)+ma(12)>0 then {
2830        symbol(430,cj*40+226,str$(ma(10)+ma(11)+ma(12)),1,1,1
,15,0)
2840        symbol(426,cj*40+246,"-"+str$(te(10)+te(11)+te(12)),1
,1,1,15,0)
2850     }
2860     for i=0 to 12:if ma(i)>0 then mm=mm+ma(i)
2870     next
2880     symbol(476,cj*40+226,str$(mm),1,1,1,15,0)
2890 endfunc
2900 /*
2910 /* complay
2920 func com(cj)
2930     int i,ii,k
2940     fill(188,176,256,192,0):symbol(188,176,nam(cj)+"の番",1,
1,1,1,0)
2950     box(5,cj*100+110,43,cj*100+130,15)
2960     for i=0 to m(cj)-2
2970        for ii=i+1 to m(cj)-1
2980           if c(cj,i)<c(cj,ii) then {
2990              k=c(cj,i):c(cj,i)=c(cj,ii):c(cj,ii)=k}
3000        next   /*数の大きいカードから出すための並べ変え
3010     next
3020     if cc(m(5)-1)>100 then etcom(cj) else uscom(cj)
3030     wait(40)
3040     symbol(8,cj*100+112,nam(cj),1,1,1,15,0)
3050     box(5,cj*100+110,43,cj*100+130,0)
3060 endfunc
3070 /* your play
3080 func you()
3090     int i,x,y,a,b,yc,a1,a2,b1,b2,s=0
3100     fill(188,176,256,192,0):symbol(188,176,nam(3)+"の番",1,1
,1,1,0)
3110     box(5,426,43,446,15)
3120     symbol(176,16,"出したいカードを",1,1,1,15,0)
3130     symbol(176,35,"マウスでクリック",1,1,1,15,0)
3140     box(178,58,302,136,1):line(179,80,301,80,1):line(210,59,
210,135,1)
3150     symbol(184,61,"左    出    す",1,1,1,15,0)
3160     symbol(232,83,"ストック",1,1,1,15,0)
3170     symbol(184,100,"右     をめくる",1,1,1,15,0)
3180     symbol(232,117,"又はパス",1,1,1,15,0)
3190     mouse(4)
3200     mouse(1)
3210     msarea(49,401,495,495)
3220     setmspos(56,435)
3230     repeat
3240        msstat(x,y,a,b)
3250     until a=-1 or b=-1
3260     mspos(x,y)
3270     mouse(0)
3280     erupms()
3290     if m(3)<10 then {
3300        i=(x-48) ¥ 50:if i>m(3)-1 then i=m(3)-1
3310     } else if m(3)>9 and m(3)<18 then {
3320        if x>(m(3)-1)*25+48 then{
3330              i=m(3)-1
3340        } else i=(x-48) ¥ 25
3350     } else if m(3)>17 then {
3360        if x>(m(3)-1)*16+48 then {
3370           i=m(3)-1
3380        } else i=(x-48) ¥ 16
3390     }
3400     a1=(c(3,i)-1) ¥ 13:a2=(c(3,i)-1) mod 13
3410     if cc(m(5)-1)>100 then {
3420        b1=cc(m(5)-1)-110:b2=100
3430     } else {
3440        b1=(cc(m(5)-1)-1) ¥ 13:b2=(cc(m(5)-1)-1) mod 13
3450     }
3460     if a=-1 then {
3470        if a2=7 or a1=b1 or a2=b2 then {
3480           yc=c(3,i):c(3,i)=0:cdleft(3,i):dsyou(yc)
3490           wait(20):fill(256,176,352,192,0)
3500        } else dame1():you()
3510     } else if b=-1 then {
3520        for j=0 to m(3)-1
3530           a1=(c(3,j)-1) ¥ 13:a2=(c(3,j)-1) mod 13
3540           if a1=b1 or a2=b2 then s=s+1
3550        next
3560        if s>0 then dame2():you() else nocd(3):if g=1 then g=
0:play() else if pa=0 then you()
3570     }
3580     symbol(8,428,nam(3),1,1,1,15,0):box(5,426,43,446,0)
3590     jj=0
3600 endfunc
3610 /* subroutine of you()
3620 func dsyou(ba)
3630     int x,y,a,b,sitei
3640     fill(176,348,368,364,0)
3650     m(3)=m(3)-1:if m(3)=0 then g=1
3660     cc(m(5))=ba:m(5)=m(5)+1
3670     plcd():pas=0
3680     bacd(ba,m(5))
3690     if (ba-1) mod 13 = 7 and g<>1 then {
3700        fill(256,176,352,192,0)
3710        symbol(256,176,"8が出ました",1,1,1,5,0)
3720        m_play(3)
3730        erupms():symbol(200,12,"指定したい",1,1,1,15,0)
3740        symbol(176,32,"マークをクリック",1,1,1,15,0)
3750        fill(200,55,280,73,15):symbol(208,56,suit(0),1,1,1,1,
0)
3760        fill(200,75,280,93,15):symbol(216,76,suit(1),1,1,1,5,
0)
3770        fill(200,95,280,113,15):symbol(216,96,suit(2),1,1,1,5
,0)
3780        fill(200,115,280,133,15):symbol(216,116,suit(3),1,1,1
,1,0)
3790        mouse(1)
3800        msarea(208,56,272,132)
3810        setmspos(246,63)
3820        repeat
3830           msstat(x,y,a,b)
3840        until a=-1 or b=-1
3850        mspos(x,y)
3860        mouse(0)
3870        if y>55 and y<73 then {
3880           sitei=0:fill(200,55,280,73,1)
3890           symbol(208,56,suit(0),1,1,1,15,0)
3900        } else if y>75 and y<93 then {
3910           sitei=1:fill(200,75,280,93,5)
3920           symbol(216,76,suit(1),1,1,1,15,0)
3930        } else if y>95 and y<113 then {
3940           sitei=2:fill(200,95,280,113,5)
3950           symbol(216,96,suit(2),1,1,1,15,0)
3960        } else {
3970           sitei=3:fill(200,115,280,133,1)
3980           symbol(216,116,suit(3),1,1,1,15,0)
3990        }
4000        symbol(176,348,suit(sitei)+"指定",2,1,1,15,0)
4010        switch sitei
4020           case 0 : cc(m(5)-1)=110:break
4030           case 1 : cc(m(5)-1)=111:break
4040           case 2 : cc(m(5)-1)=112:break
4050           case 3 : cc(m(5)-1)=113
4060        endswitch
4070        wait(50):erupms():fill(256,175,352,192,0)
4080     } else fill(256,168,352,192,0):symbol(268,176,"出します"
,1,1,1,1,0)
4090 endfunc
4100 /* subroutine of you()
4110 func dame1()
4120     symbol(180,48,"出せません",1,1,2,5,0)
4130     m_play(4):wait(50):erupms()
4140 endfunc
4150 /* subroutine of you()
4160 func dame2()
4170     symbol(192,48,"出せます",1,1,2,5,0)
4180     m_play(5):wait(50):erupms()
4190 endfunc
4200 /* subroutine of com(cj)
4210 func uscom(cj)
4220     int i,a1,a2,b1,b2,s=0,et=0,sb=0,pb=0
4230     b1=(cc(m(5)-1)-1) mod 13:b2=(cc(m(5)-1)-1) ¥ 13
4240     for i=0 to m(cj)-1
4250        a1=(c(cj,i)-1) mod 13:a2=(c(cj,i)-1) ¥ 13
4260        if a1=7 then et=et+1        /* 8 の手持ちの枚数
4270        if a1<>7 and a1=b1 then pb=pb+1 /* 数字の一致するカー
ドの枚数
4280        if a1<>7 and a2=b2 then sb=sb+1 /* マークの一致するカ
ードの枚数
4290     next
4300     if pb>0 then {
4310        for i=0 to m(cj)-1
4320           a1=(c(cj,i)-1) mod 13
4330           if a1<>7 and a1=b1 then {
4340              s=1:jj=jj+1:dsuscd(cj,c(cj,i))
4350              c(cj,i)=0:cdleft(cj,i):break
4360           } else s=0  /* 8以外の数の合うカードを出す
4370        next
4380     } else if sb>0 then {
4390        for i=0 to m(cj)-1
4400           a1=(c(cj,i)-1) mod 13:a2=(c(cj,i)-1) ¥ 13
4410           if a1<>7 and a2=b2 then {
4420              s=1:jj=jj+1:dsuscd(cj,c(cj,i))
4430              c(cj,i)=0:cdleft(cj,i):break
4440           } else s=0  /* 8以外のマークの合うカードを出す
4450        next
4460     } else if et>0 and (rnd()*9>3 or pas>1 or m(cj)<3) then
{
4470        for i=0 to m(cj)-1
4480           a1=(c(cj,i)-1) mod 13
4490           if a1=7 then {
4500              s=1:jj=jj+1:ds8cd(cj,c(cj,i))
4510              c(cj,i)=0:cdleft(cj,i):break
4520           } else s=0    /* 8 を出す
4530        next
4540     } else if s=0 then nocd(cj):if g=1 then g=0:play() else
if pa=0 then uscom(cj)
4550 endfunc
4560 /*
4570 func cdleft(j,k)
4580     int i
4590     for i=0 to m(j)-k:c(j,k+i)=c(j,k+i+1):next
4600 endfunc
4610 /* subroutine of com(c,j)
4620 func dsuscd(j,ba)
4630     fill(256,176,352,192,0):symbol(268,176,"出します",1,1,1,
1,0)
4640     fill(176,348,368,364,0)
4650     m(j)=m(j)-1:if m(j)=0 then g=1
4660     cc(m(5))=ba:m(5)=m(5)+1
4670     min_com(j,m(j)):pas=0
4680     bacd(ba,m(5))
4690     wait(10):fill(256,176,352,192,0)
4700 endfunc
4710 /* subrourine of com(cj)
4720 func ds8cd(j,ba)
4730     int i,ii,k
4740     dim int ka(3),su(3)
4750     m(j)=m(j)-1:if m(j)=0 then g=1
4760     cc(m(5))=ba:m(5)=m(5)+1
4770     min_com(j,m(j)):pas=0
4780     bacd(ba,m(5))
4790     fill(176,348,368,364,0)
4800     for i=0 to 3:su(i)=0:next
4810     if g<>1 then {
4820        fill(256,176,352,192,0):symbol(256,176,"8が出ました"
,1,1,1,5,0)
4830        m_play(3)
4840        for i=0 to m(j)-1
4850           if (c(j,i)-1)¥13=0 then su(0)=su(0)+1
4860           if (c(j,i)-1)¥13=1 then su(1)=su(1)+1
4870           if (c(j,i)-1)¥13=2 then su(2)=su(2)+1
4880           if (c(j,i)-1)¥13=3 then su(3)=su(3)+1
4890        next
```

▶ついにMIDIを買ってしまった。あまりバイトするもんじゃないなあ。金づかいがあらくなってしまう。この頃自分がこわくなってきた。　　山本　盛雄(16)滋賀県

```
4900        for i=0 to 3:ka(i)=su(i):next
4910        for i=0 to 2:for ii=i+1 to 3
4920        if su(i)<su(ii) then k=su(i):su(i)=su(ii):su(ii)=k
4930        next:next   /* 指定するマークを決定
4940        wait(40)
4950        for i=0 to 3:if su(0)=ka(i) then {
4960           symbol(176,348,suit(i)+"指定",2,1,1,15,0)
4970           switch i
4980              case 0 : cc(m(5)-1)=110:break
4990              case 1 : cc(m(5)-1)=111:break
5000              case 2 : cc(m(5)-1)=112:break
5010              case 3 : cc(m(5)-1)=113
5020           endswitch
5030           break}
5040        next
5050     }
5060 endfunc
5070 /* subroutine of com(cj)
5080 func etcom(cj)
5090     int i,s=0
5100     if cc(m(5)-1)>100 then cc(m(5)-1)=cc(m(5)-1)-110
5110     for i=0 to m(cj)-1
5120        if (c(cj,i)-1)¥13=cc(m(5)-1) and (c(cj,i)-1) mod 13=7
 then {
5130           s=1:jj=jj+1:ds8cd(cj,c(cj,i))
5140           c(cj,i)=0:cdleft(cj,i):break
5150        } else if (c(cj,i)-1)¥13=cc(m(5)-1) and (c(cj,i)-1) m
od 13<>7 then {
5160           s=1:jj=jj+1:dsuscd(cj,c(cj,i))
5170           c(cj,i)=0:cdleft(cj,i):break
5180        } else s=0
5190     next
5200     if s=0 then nocd(cj):if g=1 then g=0:play() else if pa=0
 then etcom(cj)
5210 endfunc
5220 /* case of no applicable card
5230 func nocd(cj)
5240     pa=0
5250     if m(4)>0 then {
5260        fill(256,176,352,192,0):symbol(268,176,"取ります",1,1
,1,1,0)
5270        wait(30):fill(256,176,352,192,0)
5280        m_play(6):m(4)=m(4)-1:if m(4)<=0 then m(4)=0
5290        m(cj)=m(cj)+1: if m(cj)>26 then nocount()
5300        c(cj,(m(cj)-1))=s(m(4))
5310        min_st(m(4)):if cj=3 then plcd() else plus_com(cj,m(c
j))
5320        wait(40)
5330        erupms()
5340     } else if m(4)=0 then {
5350        fill(256,176,352,192,0):symbol(268,176,"パスします",1
,1,1,5,0)
5360        wait(30):fill(256,176,352,192,0)
5370        m_play(7):pa=1:pas=pas+1:fill(176,348,368,364,0)
5380        if pas>5 then nocount()
5390     }
5400 endfunc
5410 /* case of nocount
5420 func nocount()
5430     fill(184,176,352,192,0)
5440     symbol(184,176,"ノーカウント やり直し",1,1,1,5,0):m_play
(4)
5450     g=1
5460 endfunc
5470 /* make ba
5480 func mkba()
5490     int i
5500     apage(2)
5510     fill(161,145,375,383,8):fill(176,168,360,200,15)
5520     fill(377,145,509,383,10):symbol(380,168,"ストック カー
ド",1,2,1,15,0)
5530     fill(2,81,159,383,10)
5540     for i=0 to 2:symbol(8,i*100+112,nam(i),1,1,1,15,0):next
5550     apage(1)
5560 endfunc
5570 /* select
5580 func sel(x,y,m,n)
5590     int i,j,a,b
5600     str mm,nn
5610     switch m
5620        case 1:mm="必  要":break
5630        case 2:mm="O  K"
5640     endswitch
5650     switch n
5660        case 1:nn="不  要":break
5670        case 2:nn="やめる"
5680     endswitch
5690     fill(x,y,x+56,y+24,15):fill(x+72,y,x+128,y+24,15)
5700     symbol(x+4,y+4,mm,1,1,1,1,0):symbol(x+76,y+4,nn,1,1,1,1,
0)
5710     mouse(1)
5720     msarea(x+1,y+1,x+127,y+23)
5730     setmspos(x+28,y+8)
5740     repeat
5750        msstat(i,j,a,b)
5760     until a<>0 or b<>0
5770     mspos(i,j)
5780     mouse(0)
5790     if i<x+64 then {
5800        fill(x,y,x+56,y+24,1):symbol(x+4,y+4,mm,1,1,1,15,0):p
=1
5810     } else {
5820        fill(x+72,y,x+128,y+24,1):symbol(x+76,y+4,nn,1,1,1,15
,0):p=2
5830     }
5840     return(p):wait(40)
5850 endfunc
5860 /* click
5870 func click()
5880     int i,j,a,b
5890     symbol(176,48,"よければマウスを",1,1,1,15,0)
5900     symbol(208,84,"クリック",1,1,1,15,0)
5910     mouse(1)
5920     msarea(176,48,288,96)
5930     setmspos(232,70)
5940     repeat
5950        msstat(i,j,a,b)
5960     until a<>0 or b<>0
5970     mouse(0)
5980     erupms()
5990 endfunc
6000 /* player's cards
6010 func plcd()
6020     int i
6030     fill(48,400,496,496,0)
6040     if m(3)<10 then {
6050        for i=0 to m(3)-1
6060           c_put(i*50+48,400,c(3,i))
6070           m_play(1,2)
6080        next
6090     } else if m(3)>9 and m(3)<18 then {
6100        for i=0 to m(3)-1
6110           c_put(i*25+48,400,c(3,i))
6120           line(i*25+47,400,i*25+47,496,10)
6130           m_play(1,2)
6140        next
6150     } else if m(3)>17 then {
6160        for i=0 to m(3)-1
6170           c_put(i*16+48,400,c(3,i))
6180           line(i*16+47,400,i*16+47,496,10)
6190           m_play(1,2)
6200        next
6210     }
6220     fill(16,453,32,469,0)
6230     symbol(16,453,str$(m(3)),1,1,1,15,0)
6240 endfunc
6250 /* com's card
6260 func comcd(c)
6270     int i
6280     fill(48,c*100+84,146,c*100+180,0)
6290     for i=0 to m(c)-1
6300        c_put(i*2+48,c*100+84,0)
6310        line(i*2+47,c*100+84,i*2+47,c*100+180,10)
6320        m_play(1,2)
6330     next
6340     fill(16,c*100+137,32,c*100+153,0)
6350     symbol(16,c*100+137,str$(m(c)),1,1,1,15,0)
6360 endfunc
6370 /* minus com cards
6380 func min_com(c,mai)
6390     fill((mai+1)*2+46,c*100+84,(mai+1)*2+94,c*100+180,0)
6400     if mai>0 then c_put(mai*2+46,c*100+84,0)
6410     m_play(1,2)
6420     fill(16,c*100+137,32,c*100+153,0)
6430     symbol(16,c*100+137,str$(mai),1,1,1,15,0)
6440 endfunc
6450 /* plus com cards
6460 func plus_com(c,mai)
6470     c_put(mai*2+46,c*100+84,0)
6480     line(mai*2+45,c*100+84,mai*2+45,c*100+180,10):m_play(1,2
)
6490     fill(16,c*100+137,32,c*100+153,0)
6500     symbol(16,c*100+137,str$(mai),1,1,1,15,0)
6510 endfunc
6520 /* stock card
6530 func stcd()
6540     int i
6550     for i=0 to m(4)-1
6560        c_put(i*2+388,240,0)
6570        line(i*2+387,240,i*2+387,336,10)
6580        m_play(1,2)
6590     next
6600     fill(440,208,456,224,0)
6610     symbol(440,208,str$(m(4)),1,1,1,15,0)
6620 endfunc
6630 /* minus stock card
6640 func min_st(mai)
6650     fill((mai+1)*2+386,240,(mai+1)*2+434,336,0)
6660     if mai>0 then c_put(mai*2+386,240,0):m_play(1,2)
6670     fill(440,208,456,224,0)
6680     symbol(440,208,str$(mai),1,1,1,15,0)
6690 endfunc
6700 /* ba card
6710 func bacd(ba,mai)
6720     c_put(mai*3+165,240,ba):line(mai*3+165,240,mai*3+165,336
,8)
6730     if mai>1 and (ba-1) mod 13 = 7 then box(mai*3+165,240,ma
i*3+213,336,5)
6740     m_play(1,2)
6750     fill(260,208,276,224,0)
6760     symbol(260,208,str$(mai),1,1,1,15,0)
6770 endfunc
6780 /*
6790 func wait(t)
6800     int i
6810     for i=0 to t*100:next    /*C に変換する時は t*200 とする
6820 endfunc
6830 /*
6840 func erupms()
6850     fill(161,3,319,143,0)
6860 endfunc
6870 /* rule
6880 func rule()
6890     apage(0)
6900     line(2,144,159,144,15):fill(2,145,509,383,12)
6910     symbol(196,160,"ル  ー  ル",1,1,1,15,0)
6920     symbol(60,192,"1 : 最初の手持ちは5枚、早く無くなった人
が勝ち",1,1,1,15,0)
6930     symbol(60,212,"2 : 同じスーツ又は同じ数字なら出せます",1
,1,1,15,0)
6940     symbol(60,232,"3 : 8は何時でも出せます 但し次の人のカ
ードを指定する",1,1,1,15,0)
6950     symbol(60,252,"4 : 出すカードがない時はストックから取り
ます",1,1,1,15,0)
6960     symbol(120,270,"ストックが無くなればパスします",1,1,1,15
,0)
6970     symbol(60,290,"5 : 誰かが上がれば終り",1,1,1,15,0)
6980     symbol(120,308,"残ったカードが各々のマイナス点になります
",1,1,1,15,0)
6990     symbol(60,328,"6 : 8は50点, Aは1点, J,Q,Kは10点, 他は
そのまま",1,1,1,15,0)
7000     apage(1)
7010 endfunc
```

X68000CARDDRV用カードゲーム

コンパイラ対応カードゲーム変更点

Mounai Toshiyuki 毛内 俊行

CARDDRVによってコンパイラ対応となったカードゲームシステム。そこでこれまでに発表されたカードゲームをすべてコンパイルできるようにしてみました。加えて，キーボートでしかできなかったゲームをマウス対応にする変更点も掲載します。

コンパイラユーザーの方へ

皆さんこんにちは，CARD2.FNCは使っていますか？　1月号の付録ディスクには，CARD2.FNCのほかにコンパイラ用のライブラリが入っていたので，さっそく使ってみた人も多いのではないかと思います。

ところが実際にコンパイルして見ると，いままでX-BASIC（インタプリタ）ではちゃんと動いていたプログラムがまったく動かなかったり，コンパイルできなかったりして，皆さんの頭を悩ませたのではないでしょうか？　なかには「ライブラリのバグでは？」と考えた人もいることでしょう。私も実際にコンパイルしてみて，動かないプログラムがあまりにも多いのに驚いてしまいました。でも，これはライブラリが悪いのではありません。これらの原因はすべて，ゲームのプログラムに責任があるのです。

コンパイラを使い慣れた人ならわかることなのですが，X-BASICとそのコンパイラを比較した場合，X-BASICではプログラムの中でちょっとルール違反をした場合でも「しょうがないなあ」と黙認してくれたりします。

しかしコンパイラはそうはいきません。コンパイラはそのプログラムをマシン語に変換するわけです。もし，マシン語に変換したプログラムがルール違反をしたままだったら……。そうです，バスエラーなどの原因となったり，まかり間違えばシステムを破壊することだってあるかもしれません。そんなわけでコンパイルするプログラムというのは，とても礼儀正しくしていなければいけないのです。

そこで今回はコンパイラユーザー必見，「過去に発表された全カードゲームのコンパイル方法」と題しまして，1990年5月号～12月号で発表された全ゲームのリストを調べ，礼儀正しくないところを指摘してコンパイルできるプログラムに改造する方法を発表しようと思います。

なお，今回はコンパイラにXCver2.0を使用し，最適化などのオプションは一切つけずにコンパイルしました。おそらくどんなオプションをつけても大丈夫だとは思いますが，プログラムの調子が悪いときはオプションなしでコンパイルしてみてください。

変更プログラム

各ゲームの変更点をリスト1にまとめました。X-BASICを起動してから，各ゲームのプログラムをロードして，該当する箇所のプログラムを書き換え，セーブしてください。あとはX-BASICを終了し，コマンドラインから，

CC　ファイル名　CARDLIB.A

と実行するだけです。

リスト1-HのKlondikeのソースプログラムは，1月号の付録ディスク3のCARD2というディレクトリに格納されています。Klondikeは，付録ディスクに収められている実行ファイルに少し不備がありましたので，コンパイラを持っている人は変更箇所を入力し，もう一度コンパイルし直してください。なお，従来のKlondikeでもゲーム自体には特に支障はないのでコンパイラを持っていない方は，付録ディスクに収められているKlondike.xをそのまま使って大丈夫です。

なお，おまけとして99とローリングストーンの2つのゲームをマウスに対応させるプログラム（変更点）も用意しました。よろしければ使ってください。リスト2とリスト3がそれです。なおこの場合、1ゲーム終了するたびにマウスの左ボタンで次のゲームへ進み、右ボタンでゲームが終了するようになっています。ほかのゲームも、ブレイクキーを押さなければ終わらなかったゲームはすべてこのように改造してあります。

99やローリングストーンをマウス対応に改造しない場合は、キー入力待ちのときにブレイクキーを押せば（1発できかなくても2、3回押せばきく）ゲームを終了できるはずです。

さて、それではなぜ、これらのゲームがコンパイルできなかったのか、これからじっくり検証していこうと思います。

コンパイルできなかった理由

では本題です。今回コンパイラ対応に改造したプログラム群を調べた結果からいうと，大まかに分けて，コンパイルしたゲームに問題が発生した場合，それは次の3種類に分類されます。

1)　まともに動くが，速度が速くなった影響でゲームバランスが乱れた。

2)　ゲームをコンパイル，起動できたが，動作がおかしい。バスエラーが発生する場合もある。

3)　コンパイルできなかった。

1）の場合はたいして問題はありません。コンパイルした影響でゲームが速くなってしまったというだけなら，プログラム自体に間違いは見当たらないのですから。それなら，プログラムの適当なところでウエイトをかけてやれば解決するでしょう。99やブラックジャックがこの例です。

また速度が速くなったために，思わぬ動

作をした例もあります。たとえばCOUPLEなどは，マウスボタンが押しっぱなしになっていても検出できなかったので，マウスボタンをちょこんと押しただけで「しゅぱぱぱぱ」とカードが何枚も表示されてしまい，ゲームになりませんでした。この場合，プログラム中のどこかにマウスボタンをOFFにしたことを確認するプログラムを付け足せば，このような現象を回避することができます。

2)の場合は少々厄介なので先に3)の場合から説明します。この場合は，コンパイルしている最中にエラーを発生させて止まってしまったのですから，文法や構造など，プログラムに明らかに間違いがあると考えていいでしょう。今回はHEARTとローリングストーンのリスト中にその間違いを見ることができます。

HEARTの場合，900行のsymbol文のストリング引数“及び”の後ろに区切りのコンマがありません。なんとX-BASICでは，関数の引数の区切りが明らかなときは，コンマを省略してもエラーにならないのです。これこそ「小さな親切，大きなお世話」でしょう。

ローリングストーンの場合，3520行からのif文の“}”をつけ忘れています。普通，if文を閉じ忘れればX-BASICでもエラーになるのですが，この場合はreturn文でif文を閉じる前に関数を終了しているので，インタプリタが実行している限りエラーにはならなかったわけです。このような場合は文法を間違えた箇所を訂正してやればいいので，簡単に修正ができるでしょう。

さて，それでは2)のような場合はどうしたらいいのでしょうか。コンパイラはエラーを出さなかったのだから，文法に間違いはないと考えていいはずです。しかしプログラムはまともに実行できない。一見目に見えない，こんなバグが一番厄介なのです。

しかしたいていの場合，これは変数の初期化を忘れたために発生すると考えていいでしょう。実はほかの原因も十分に考えられるのですが，発生する頻度からいって(特にX-BASICからコンパイルした場合)このバグが発生する確率が高くなります。

X-BASICで変数を定義するには，

1000 int A

のように書くということは誰もが知ってい

リスト1

```
1－A

==================  COUPLE.bas  ==================
 175 c_palet()
 640     msstat(dm,dm,bl,br)
 650   until bl+br<>0
 655   if br=-1 then color 3:width 96:end
 721       repeat
 722         msstat(dm,dm,bl,br)
 723       until bl+br=0
1740   for i=1 to 384:home(0,0,i):wait(200):next
1830   for i=385 to 896:home(0,0,i):wait(200):next


1－B

==================  BlackJack.bas  ==================
 590     kput(i,j,1):wtm(1)
 700 for i=0 to n*1500:next
2895 if r=-1 then width 96:end


1－C

==================  HEART.BAS  ==================
 900                 symbol(224,56,"及び",1,1,1,15,0)
3130           int i,j,a,b,cla=0


1－D

==================  AKAGURO.BAS  ==================
 341   repeat:msstat(x,y,lb,rb):until lb+rb=0
1180   int w,x,y
1215   x=lr*121+172+rnd()*10-5:y=208+rnd()-5
1220   c_put(x,y,c_num)
1450   int i, j, rs=0, bs=0, rc=0, bc=0
2755   int dm,ql,qr
2756   repeat:msstat(dm,dm,ql,qr):until ql+qr=0


1－E

==================  TEN.bas  ==================
  80 palet(1,0):mouse(4):/*ｶｲﾘｮｳ
1505           repeat:msstat(x,y,bl,br):until bl+br=0
1715           repeat:msstat(x,y,bl,br):until bl+br=0
1760  /*       mouse(0):/*ｶｲﾘｮｳ
1780           xx=(x-16)¥64:yy=(y-183)¥104:i=yy*5+xx
1790           if i=14 then i=15
1800           if i=10 then i=20
1810           xx=xx*64+15:yy=yy*104+183
1820           if i<15 then box(xx,yy,xx+48,yy+97,5)
1830           if i>9 and i<14 then i=i-1
1840           return(i)
1850 endfunc
1860 /*
1870 /*
1880 /*


1－F

==================  99.BAS  ==================
 110 int PN,PV=1,BA=0,BP,TMR=1500,CX=0,YS=0


1－G

==================  RollingStone.bas  ==================
  90 int i,fpl=0,tmr=1000,cy=0,demo=0,win
1685 for i=0 to 31:a(i)=0:next
3560   return(drop21(pl,i))}
3965 for i=0 to 31:hbf(i)=0:next


1－H

==================  KLONDIKE.BAS  ==================
4750   int ow1,ow2=0,i,ms,x,y,t1,t2,t3,t4
```

ると思います。それではこの場合，変数Aを定義した直後，その変数の中に格納されている数はいくつだかわかるでしょうか？
「そんなの0に決まっているよ」
そう答えた人はあまい！　X-BASICでは確かに便宜上0が格納されますが，コンパイルしたプログラムでは，変数の中に格納されている数は0とは限りません(特にローカル変数に注意。グローバル変数は大丈夫なはず)。つまり，このようなプログラムがあった場合必要なら，

1000 int A=0

と初期化してやる必要があるのです。
このバグは，実行して動かなかったプログラムのほとんどに見つかりました。CARD2.FNC用のゲームに限らず，X-BASICのプログラムををコンパイルした場合には，必ず確認しなければいけないミスでしょう。
このほかに，TENではプログラムそのものにバグが潜んでいました。X-BASIC上で使っている分には，たいした誤動作もしなかったので発見できなかったのですが，コンパイルしてみてときどき不可解なバスエラーを発生させるので調べてみたら，バグが発見されたというわけです。そのためTENでは1780行からのプログラムを新しく書き直すことで対処しました。

リスト2

```
==================  99.BAS  ==================
 140 mouse(1):mouse(4)
 160 INIT():msarea(16,490,207,491)
 280 width 64
 290 locate 20,10:print"役札の説明が必要ですか？"
 300 locate 20,12:print"  [Yes]          [No]"
 310 YS=YESNO()
 320 /*
1250 int A,X,mx,my,bl,br
1260 mspos(mx,my):X=(mx-16)¥48
1270 fill(X*48+16,490,X*48+64,493,6)
1280 repeat
1290   msstat(mx,my,bl,br):mspos(mx,my):X=(mx-16)¥48
1300   if X<>CX then fill(16,490,208,493,0)
1310   fill(X*48+16,490,X*48+64,493,6)
1320   CX=X
1330 until bl+br<>0
1340 fill(16,490,208,493,0)
1350 return(X)
1360 endfunc
1370 /*
1380 /*
1410 int A,Y,mx,my,bl,br
1420 msarea(7,11,8,479)
1430 mspos(mx,my):Y=(my-11)¥96
1440 repeat:msstat(mx,my,bl,br):until bl+br=0
1450 repeat
1460   msstat(mx,my,bl,br):mspos(mx,my)
1470   A=(my-11)¥96
1480   if A<>Y then Y=A:NM_MARK(Y)
1490 until bl+br<>0
1500 msarea(16,490,207,491)
1510 return(Y)
1520 endfunc
3160 int X,Y,mx,my,bl,br
3210 MSG(36,28," click left button!!")
3280 repeat:msstat(mx,my,bl,br):until bl+br=0
3290 repeat:msstat(mx,my,bl,br):until bl+br<>0
3300 if br=-1 then mouse(0):end else BA=0:PV=1
3510 func YESNO()
3520 int mx,my,bl,br
3530 setmspos(247,200)
3540 while 1
3550   repeat:msstat(mx,my,bl,br):until bl+br=0
3560   repeat:msstat(mx,my,bl,br):until bl+br<>0
3570   mspos(mx,my)
3580   if my>208 or my<192 then continue
3590   if mx>175 and mx<215 then return(1)
3600   if mx>287 and mx<319 then return(0)
3610 endwhile
3620 return(0)
3630 endfunc
```

リスト3

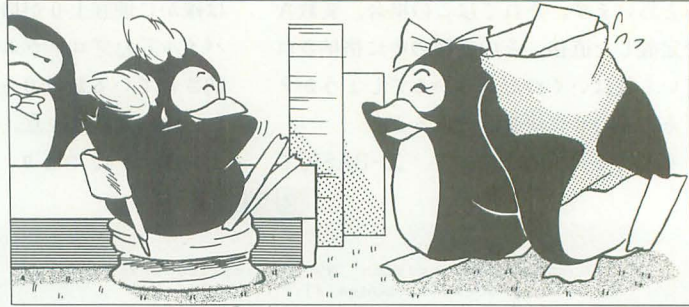
```
==================  RollingStone.bas  ==================
 140 dinit():mouse(1):mouse(4):msarea(16,470,321,471)
 305 repeat:msstat(x,y,l,r):until l+r=0
 310 repeat:msstat(x,y,l,r):until l+r<>0
 315 if r=-1 then mouse(0):end
2740 msarea(16,470,(k+1)*sp+15,471)
2750 fill(16,470,400,473,0)
2760 fill(cy*sp+16,470,(cy+1)*sp+15,473,4)
2770 repeat:msstat(x,y,l,r):until l+r=0
2780 repeat
2790   msstat(x,y,l,r):mspos(x,y)
2800   k=(x-16)¥sp
2810   if k<>cy then fill(16,470,400,473,0)
2820   cy=k
2830   fill(cy*sp+16,470,(cy+1)*sp+15,473,4)
2840 until l+r<>0
2860 fill(16,470,400,473,0)
2870 return(cy)
2880 endfunc
2890 /*
```

最後に

コンパイラ用のライブラリを作ったとき，まさかこんなにコンパイルできないプログラムが多いとは思いませんでした。当然，ライブラリの制作者としては「むむっ，これはいかん！」と思うわけで，正月三賀日を使ってプログラムの改良(改造ではない)をしてしまったのです。
この記事は最初，もっと簡単に「こんなところがいけないんだよ」と説明するつもりで書き始めたのですが，結構だらだらと長くなってしまいました。いままでに発表されたカードゲームをコンパイルできるようにするだけなら，リストの変更箇所だけを載せればよかったのですが，私としては皆さんに，もう少しコンパイラのことを考えてほしいと思ったので，今回のような原因の究明をしたわけです。
プログラムを作る際，今回説明したことを頭の片隅に置いていてくれれば，それだけでコンパイルする人の労力は半分に減ることでしょうし，プログラムをコンパイルするときも，バグの原因がなんなのか見当をつけることができることでしょう。
今回の記事が，ただのプログラム変更記事ではなく，今後X-BASICのプログラムを作る人にとって役に立つことを願っています。

マシン語カクテル in Z80's Bar

第19回――限りある資源をハフマンで――

シナリオ&イラスト：山田純二
特別監修：浦川博之

今回はデータ圧縮のお話。そんなに大きなものは扱わないからあまり関係ないという人もいるでしょうが，いろんな方法や考え方は知っておいて損はないですから，しっかり読んでくださいね。しかし，長老はやっぱり体より口のほうがよく動くようですね……。

♪カラン，コローン（ドアの開く音）

マスター（以下M）：あ，いらっしゃい。

長老（以下老）：こんにちは。おや，今日はおぬしひとりだけかの。

M：ええ。

老：いよいよここも世紀末か。そのうち，借金で首の回らなくなったマスターがようこちゃんとメアリーをホンコンに売りとばして，遠い北の果てへ逃亡……。

M：んなことになるわけないでしょう！　もう，縁起でもないことをいわないでくださいよ。

老：しかし，現にいまここにいるのはおぬしだけじゃないか。ほかに客がいる気配はないし。わしら以外の客なんか存在するんじゃろうか。

M：ちゃんといますよ。それにいまは開店時間前でお客がいないのは当然です。

老：ふぉっふぉっ，冗談じゃよ。ところで，ようこちゃんとメアリーはどこへ行ったのかの。

M：彼女たちは買い物ですよ。

老：そうか。こんな閉店寸前の店のためにかいがいしく働いてくれる娘がいて，マスターも幸せ者じゃのう。

M：長老も結構しつこいですね。

老：まだまだ。で，おぬしはなにをやっておるのじゃ。

M：お客の置いていったプログラムの整理ですよ。最近，プログラムの数が増えてしまって管理が大変なんですよ。

老：あいかわらずいいかげんじゃのう。

M：そろそろ店のハードディスクも容量が少なくなってきたし，どうしようか悩んでいるんですけどなにかいい方法はないもんですかね。

老：うーん，ファイルの圧縮をするのがいちばんいいかのう。ついでに，複数のファイルをひとつにまとめるアーカイバもあれば，ずいぶん楽になると思うのじゃが。

M：なるほど。ほとんど使わないファイルなんかは，まとめて凍結しておけばディスクスペースの節約にもなるし，ファイルの管理も楽になるわけですね。しかし，そのプログラムがないことには問題はなにも解決しないわけで……。

老：な，なんじゃ。

M：長老！　ここはひとつうちのためにプログラムを作ってくれませんか。

老：いきなり何をいいだすのじゃ。老い先短い年寄りをこき使おうなどと考えるとは，おぬし，畳の上では死ねんぞ。

M：わけのわからない理屈なんかこねないで頼みますよ。

カラン，コロ～ン♪（ドアの開く音）

メアリー（以下メ）：タダイマー。

ようこ（以下Yo）：ただいま。と，長老さん，こんにちは。

M：あ，おかえりなさい。おや，後ろで荷物を抱えているのは山田君かな。

純二（以下純）：ようこさーん，この荷物どこに置けばいいんですか。

Yo：そこのテーブルに置いておいて。買い物先でちょうど会ったから，まんが本で買収して荷物運びさせてきたの。

メ：ワタシタチ，トッテモタスカリマシタ。

M：それはそれは，ごくろうさん。

老：じゃあ，わしはこれで失礼させてもらうとするか。

M：長老，逃げないでくださいよ。

純：どうしたんですか。

M：いやあ，長老にプログラムの作成をお願いしたんですが，理屈ばっかこねてなかなか承諾してくれないんですよ。

Yo：それはいけないわね。たまには長老も役に立つことを証明しなきゃ。

メ：ドンナプログラムナンデスカ。

M：実は……というわけ。

純：へえ，データ圧縮についてですか。僕も高校時代にアニメーションプログラムの作成をしたときに少しだけやったことがありますよ。

老：そうかそうか。そういうことならこの件は山田君にすべて任せることにしよう。

純：ほえ？

老：どうせひまなんじゃろう。

M：プログラムさえ作っていただければ，誰でも別にいいですけど。

老：ようし，決まった。

連長(ランレングス)法

老：山田君がやったことのある圧縮アルゴリズムというのはどういうものだい。

純：連長（ランレングス）法ですよ。

老：なるほどな，データ圧縮のいちばん基本ともいえるやつじゃな。

Yo：どういうふうにやっていくの。

老：ひと言でいえば，連続している同じデータを個数で表してやるのじゃ。具体的にいうと，

「あたたたたたたたた」

というデータは，

「あ8た」

となるのじゃ。

メ：展開スルトキニハドウスルンデスカ？

老：まず，「あ」というデータをメモリに置く。次に個数を表すコードの8がくるの

で，次のデータの「た」を8個分メモリに展開してやればよいのじゃ。

Yo：なるほどね。でも長老，個数を表すコードというのは，直接数値を書いてやればいいわけじゃないんでしょう。

老：うむ，そのとおりじゃ。もしも，データ中に1度も使われていないコードがあれば，そのコードを圧縮を表すコードにしてやればいい。圧縮を表すコードを「も」とした場合，先ほどの例はどうなるかわかるかの。

メ：「あも8た」ト，ナルンジャナイノデスカ。

老：そうじゃ。でも，気をつけなくてはならないのは，この圧縮は3文字以上の場合にのみ適応してやらなければならないということじゃ。

M：2文字の場合には，圧縮するどころか逆に1バイト増えてしまいますからね。

Yo：じゃあ，データ中ですべてのコードが使われていた場合はどうするんですか。

老：山田君はどうしたかの。

純：僕は，圧縮されている部分と非圧縮部分の個数データに1ビットのフラグをつけて，判断するようにしました。

メ：非圧縮部分ニモフラグヲツケテヤルンデスカ。

純：うん，そうしてやらないとどこが圧縮されているか，そうでないかがメチャクチャになってしまうからね。

Yo：圧縮効率はどれぐらいなの。

老：アルゴリズムの性格上，テキストファイルやオブジェクトファイルなどにはほとんど効果はない。

Yo：それじゃあ，あんまり使えないんじゃないの。

老：そんなことはないぞ。単純な画像データなどにはそれなりの効果があるし，プログラム次第で，高速化もやりやすい。

M：それになんといってもわかりやすいですからね。

老：そのとおりじゃ。

スライド辞書法

Yo：ねえ長老，テキストファイルやオブジェクトファイルを効率よく圧縮できるようなアルゴリズムはないんですか。

老：スライド辞書法というものがあるぞ。やり方としてはランレングス法に近く，ランレングス法では連続している文字だけ符号化していたものを，文字列にあてはめて符号化していくというものじゃ。

メ：ドウイウコトデスカ。

老：つまり，同じ文字列が再び現れた場合に，再び現れた文字列を何文字前に戻って何文字分コピーせよ，というふうに符号化してやるのじゃ。たとえば，

　　「くるくるぴかりんくるくる」

というデータは，

　　「くるくるぴかりん８４」

というふうにするのじゃ。最後から2番目の「8」は，8文字前に戻れということを意味し，「4」というのは4文字分コピーせよ，ということを意味しているのじゃ。

純：長老，「ふぉっふぉっふぉっ」というデータはどういうふうに符号化されるんですか。

老：それはじゃな，「ふぉっ３６」となるのじゃ。

メ：アレ，「ふぉっ３３６３」デハナインデスカ。

老：違うんじゃよ。

Yo：ふうん。じゃあ「ふぉっ３６」はどうやって復元されるんですか。

老：まず，「ふぉっ」がメモリに出力され，次に3文字前に戻って6文字分コピーせよという命令がくる。ここで，3文字分コピーするとメモリには，「ふぉっふぉっ」と展開される。残りの3文字分は，始めにコピーした2番目の「ふぉっ」という文字列をさらにコピーするのじゃ。

Yo：ちょっとややこしい感じがするけど，うまくできているのね。

純：でも，テキストファイルなどにはかなり効果がありそうな方法ですね。

老：そういうことじゃ。

ハフマン法

M：まだいろいろと面白そうな圧縮方法がありそうなものですが，次はなんですか。

老：ハフマン法について説明しよう。

純：どういう方法ですか。

老：うむ，通常のデータは，1文字あたり8ビットで構成されているのはみんな知っているじゃろう。

一同：はい。

老：それぞれの文字の出現頻度は，ファイルによって多い文字もあれば，一度しか出現しないような文字もあるのもわかるな。

一同：はい。

老：そこで，出現頻度が多い文字には短いコードを，少ない文字には長いコードを割り当ててやる。というのがこの方法の基本的な考え方じゃ。たとえば，あるファイルで「A」という文字がほかの文字に比べて頻繁に現れるとする。「A」の文字には2ビットぐらいのコードを割り当ててやり，ほかの滅多に現れないコードには10ビットぐらいのコードを割り当ててやるのじゃ。こうすることにより，ファイルが小さくなるのはわかるじゃろう。

一同：なるほどね。

では，ハフマン法だ

M：長老，圧縮アルゴリズムの説明はこのくらいにするとして，結局どの方法を使いますか。

老：山田君はどれを使ってみたいかの。

純：ハフマン法なんか面白そうじゃないですか。

老：よかろう。なにか質問はあるかな？

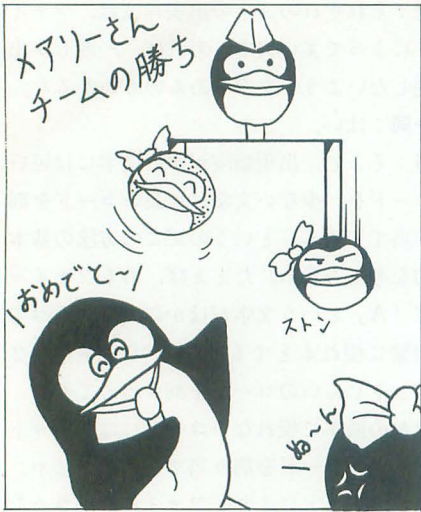
純：えっと，出現頻度に応じてそれぞれの文字にコードを割り当てるということですが，具体的にどういう方法でコードを割り当てていくんですか。

老：出現頻度を元にしてハフマン木を構成してやるのじゃ。

M：ハフマン木？

老：みんな，木構造というのは知っているじゃろう。

純：あのひとつの要素に別の要素がぶら下がって，全体で木のようなデータ構造をしているやつですね。

老：まあそんな感じじゃ。ハフマン木というのはそれぞれの出現頻度を小さい順にどんどん束ねることを繰り返して作っていくのじゃ。たとえば，文字「A, B, C, D, E」の出現比が「5：2：4：15：3」である場合，ハフマン木は図1のようになる。それぞれの文字に対応するコードは，いちばん上の根の部分から左に分岐したときにはビット0を，右に分岐したときにはビット1を出力していき文字の部分にたどりつくまでに出力されたビットがその文字のコードにな

るのじゃ（表1）。

純＆M：うーん（頭を抱える）。

老：どうしたのじゃ。

純：あのですね，ハフマン木を作成してからコードを求めるのはわかったのですが，肝心のハフマン木の作成部分がいまいちよくわからないんです。

老：じゃあ，もう一度詳しく説明するとするかの。

純：よろしくお願いします。

老：まず，AからEまでの出現頻度を集合Sとする。

S={5, 2, 4, 15, 3}

1)　この集合Sから最小の値を取り出して左に置き，その要素を集合Sから外す

S={5, 4, 15, 3}

2)　もう一度，1) と同じことをし，今度は取り出した要素を右に置く

S={5, 4, 15}

3)　取り出した左右の要素を子に持つ節点を作る。これは，左右の要素の和を値とする。この値を新たに集合Sに追加する

S={5, 4, 15, 5}

4)　以上，1) から3) までの処理を集合Sの要素がひとつになるまで続ける

純：ふうん，そうなのか。

M：実際のプログラムではどうしたらいいんですか。

老：用意すべきバッファは256×2－1の配列が5個ということになる。

純：なんで，256×2－1なんですか？

老：文字が8ビットの場合，表現できる数は256個であるのはわかるであろう。

純：はい。

老：この場合，ハフマン木を組み立てていくときに必要な節点はいくつになるか考えてごらん。

純：それぞれ，2個ずつ組み合わせてひとつの節点を作っていくのだから，511個。なるほどね。

老：わかってもらえたかな。で，5個の配列は，先ほど話した集合Sの役割を果たすHEAP[ ]，節点を格納するFREQ[ ]，節点がどの親にぶら下がっているかを示すポインタをPARENT[ ]，節点の左右に何番目の子をぶら下げているかを示すポインタをLEFT[ ]，RIGHT[ ]とする。あとは初期値256を持つカウンタ（COUNT），取り出した左右の要素が何番目かを表すLEFTN，RIGHTN，そして左右の要素の値を持つLDW，RDWの変数を用意して準備完了じゃ。

純：いよいよですね。

老：いよいよじゃ。ハフマン木を作る手順は，

1)　HEAP[ ]から最小値を取り出してLEFTN，LDWにセット

HEAP[LEFTN]=0

で，要素を消す

2)　もう一度，最小値を取り出してRIGHTN，RDWにセット

HEAP[RIGHTN]=0

で，要素を消す

3)　で，以下の処理をする。

LEFT[COUNT]=LEFTN
RIGHT[COUNT]=RIGHTN
PARENT[LEFTN]=COUNT
PARENT[RIGHTN]=COUNT

FREQ[COUNT]=RDW+LDW
HEAP[COUNT]=RDW+LDW
COUNT=COUNT+1

4)　HEAP[ ]の中の要素がひとつになるまで1) から繰り返すのじゃ。

純＆M：わ～い（パチパチパチ）。

老：ここまで説明すればプログラムは組めるじゃろう。

純：あらほらさっさ。では，さっそく家に帰ってやってみましょう。では，さらば。

カラン，コロ～ン♪

M：ありゃりゃ。あっという間に帰ってしまった。

静的がありゃ動的もある

そして数時間後……。

カラン，コロ～ン♪

純：こんにちは。長老います？

老：おや，山田君。プログラムはできたかの。

純：とりあえずできましたが，ちょっと問題があるんですよ。

老：どうしたのじゃ。

純：問題点1，ファイルの圧縮があまりうまくいかない。6Kバイトのファイルが1Kバイト弱ぐらいしか短くならないんです。問題点2，ファイルを復活させるためにはハフマン木の情報が必要なんですが，その情報をファイルに付加するとほとんど圧縮が無意味になってしまうんです。

老：なるほどな。

純：解決策はありますか。

老：問題点1については，圧縮したファイルをさらに圧縮をかけてやるか，別の方法で圧縮をかけてやってから，ハフマン圧縮をかけてやるしかないかもしれん。

問題点2については，動的ハフマン法に切り替えるのがいいかもしれん。まあ，ハフマン木の情報を短くする方法はあるのでその方法を使うかすればよいじゃろう。

純：問題点の1はしかたがないとして，2番目の動的ハフマン法とはどういうものですか？

老：実は山田君に教えたのは静的ハフマン法というやつなのじゃ。

純：動的と静的はどう違うんですか。

老：動的ハフマン法というのは，まず，全部の文字の出現頻度を1としてハフマン木を構成する。圧縮のときには，文字を取り

図1

表1

図1の場合に生成されるコード

A	011
B	000
C	010
D	1
E	001

図2

出すごとにハフマン木をたどってコード化し，取り出した文字の頻度に1を加えてハフマン木を再構成する。という動作を繰り返していくのじゃ。展開のときにも同じ操作をしてやればよい。この方法だとハフマン木の情報をファイルに付加しなくてもよいが，速度的な問題があるし，ややこしいと思ったので，君には静的ハフマン法のほうを教えたのじゃ。

純：う〜ん，そうなのか。でもプログラムを組んでしまったものはしかたがないのでハフマン木の情報を短くする方法を教えてくれませんか。

老：よかろう。うまくやれば1Kバイト弱必要なハフマン木の情報を256バイトに納めることができるのじゃ。

純：むむ，どういう方法なんですか。

老：生成されるコードを16ビットに制限して生成されたコードの長さだけを送ってやるのじゃ。これだと，512文字分で256バイトで収まるじゃろう。

純：でも，ただ単純にコード長を送ればいいってわけじゃないんでしょう。

老：そのとおりじゃ。最初のほうで説明したハフマン木を図2のようにしてやらなければならない。

純：どういう規則に従っていますか。

老：ハフマン木は左側を深くし，同じ階層では文字のコードが大きい順に左から並べていくのじゃ。

純：具体的にはどうやればいいんですか。

老：まずは，最初のとおりにハフマン木を構成する。そして，それぞれの文字のコード長を求めて，それを元に先ほど言った規則に従うようにハフマン木を再構成して，再びそれぞれの文字のコードを求めてやるのじゃ。

純：わかりました。

そういうわけだな，うん

老：ずいぶんと，長く説明してきたがようこちゃんとメアリーはどうしたんじゃ。

純：いつのまにか，僕と長老だけ？

M：そろそろ開店時間も迫ってきましたし準備が忙しいんですよ。

老：おや，いつの間にやら消えてしまったマスターではないですか。

純：まだいたんだ。

M：いるに決まってるでしょう。

Yo：マスター，店開けていいですか。

M：いいよ。長老と山田君なにか飲みます？　今日は，私のおごりですよ。

老&純：わあい，タダならなんでもいい。

つづく

参考文献：Cマガジン1991年1月号，「特集圧縮アルゴリズム入門」，奥村晴彦，吉崎栄泰

リスト1

```
0000                 1
0000                 2  ;HUFFMAN HOU
0000                 3  ; in Z80's Bar
0000                 4
1FF4 P               5  #PRINT  EQU $1FF4
1FF1 P               6  #PRINTS EQU $1FF1
1FEE P               7  #LTNL   EQU $1FEE
1FE5 P               8  #MSX    EQU $1FE5
1FE2 P               9  #MPRINT EQU $1FE2
1FD3 P              10  #GETL   EQU $1FD3
1FC1 P              11  #PRTHX  EQU $1FC1
1FBE P              12  #PRTHL  EQU $1FBE
1FB2 P              13  #HLHEX  EQU $1FB2
1FB5 P              14  #2HEX   EQU $1FB5
1FAF P              15  #WOPEN  EQU $1FAF
1FAC P              16  #WRD    EQU $1FAC
1FA6 P              17  #RDD    EQU $1FA6
1FA3 P              18  #FILE   EQU $1FA3
1FA0 P              19  #FSAME  EQU $1FA0
1F9D P              20  #FPRNT  EQU $1F9D
2006 P              21  #DIR    EQU $2006
2009 P              22  #ROPEN  EQU $2009
1F76 P              23  #KBFAD  EQU $1F76
2033 P              24  #ERROR  EQU $2033
1F70 P              25  #DTADR  EQU $1F70
1F72 P              26  #SIZE   EQU $1F72
0000                27
0000                28          OFFSET $C000-$3000
3000                29          ORG    $3000
3000                30
3000                31  EMAIN
3000 CD E2 1F       32          CALL  #MPRINT
3003 46 49 4C 45    33          DB    "FILE NAME:",00
3007 20 4E 41 4D
300B 45 3A 00
300E ED 5B 76 1F    34          LD    DE,(#KBFAD)
3012 CD D3 1F       35          CALL  #GETL
3015 1A             36          LD    A,(DE)
3016 FE 1B          37          CP    $1B
3018 C8             38          RET   Z
3019 21 0A 00       39          LD    HL,10
301C 19             40          ADD   HL,DE
301D EB             41          EX    DE,HL
301E 3E 01          42          LD    A,01
3020 CD A3 1F       43          CALL  #FILE
3023                44  HAJI2
3023 CD 09 20       45          CALL  #ROPEN
3026 DA 33 20       46          JP    C,#ERROR
3029 20 F8          47          JR    NZ,HAJI2
302B                48
302B 2A 70 1F       49          LD    HL,(#DTADR)
302E 21 00 50       50          LD    HL,$5000
3031 22 70 1F       51          LD    (#DTADR),HL
3034 22 6A 33       52          LD    (SADR),HL
3037 2A 72 1F       53          LD    HL,(#SIZE)
303A 22 6C 33       54          LD    (FSIZE),HL
303D CD A6 1F       55          CALL  #RDD
3040 DA 33 20       56          JP    C,#ERROR
3043                57
3043 2A 6A 33       58          LD    HL,(SADR)
3046 ED 5B 6C 33    59          LD    DE,(FSIZE)
304A 19             60          ADD   HL,DE
304B 22 6E 33       61          LD    (TRAN),HL
304E                62
304E CD CC 30       63          CALL  ENCODE
3051                64
3051 CD E2 1F       65          CALL  #MPRINT
3054 4C 4F 41 44    66          DB    "LOAD ADDRESS:",00
3058 20 41 44 44
305C 52 45 53 53
3060 3A 00
3062 2A 70 1F       67          LD    HL,(#DTADR)
3065 CD BE 1F       68          CALL  #PRTHL
3068 3E 2D          69          LD    A,"-"
306A CD F4 1F       70          CALL  #PRINT
306D ED 5B 72 1F    71          LD    DE,(#SIZE)
3071 19             72          ADD   HL,DE
3072 2B             73          DEC   HL
3073 CD BE 1F       74          CALL  #PRTHL
3076 CD EE 1F       75          CALL  #LTNL
3079                76
3079 CD E2 1F       77          CALL  #MPRINT
307C 45 4E 43 4F    78          DB    "ENCODE ADDRESS:",00
3080 44 45 20 41
3084 44 44 52 45
3088 53 53 3A 00
308C 2A 6E 33       79          LD    HL,(TRAN)
308F CD BE 1F       80          CALL  #PRTHL
3092 3E 2D          81          LD    A,"-"
3094 CD F4 1F       82          CALL  #PRINT
3097 2A 81 33       83          LD    HL,(CVADR)
309A CD BE 1F       84          CALL  #PRTHL
309D CD EE 1F       85          CALL  #LTNL
30A0                86
30A0 CD F8 32       87          CALL  DECODE
30A3 CD E2 1F       88          CALL  #MPRINT
30A6 44 45 43 4F    89          DB    "DECODE ADDRESS:",00
30AA 44 45 20 41
30AE 44 44 52 45
30B2 53 53 3A 00
30B6 2A 6A 33       90          LD    HL,(SADR)
30B9 CD BE 1F       91          CALL  #PRTHL
30BC 3E 2D          92          LD    A,"-"
30BE CD F4 1F       93          CALL  #PRINT
30C1 2A 74 33       94          LD    HL,(SADRW)
30C4 2B             95          DEC   HL
30C5 CD BE 1F       96          CALL  #PRTHL
30C8 CD EE 1F       97          CALL  #LTNL
30CB C9             98          RET
30CC                99
30CC               100
30CC               101  ;ENCODE MAIN
30CC               102
30CC               103  ENCODE
30CC CD E4 30      104          CALL  CODECNT
30CF 11 87 43      105          LD    DE,HEAP
30D2 21 87 33      106          LD    HL,FREQ
30D5 01 00 04      107          LD    BC,1024
30D8 ED B0         108          LDIR
30DA               109
30DA CD 11 31      110          CALL  MAKETREE
30DD CD EB 31      111          CALL  CODEGEN
30E0 CD 7B 32      112          CALL  DATACNV
30E3 C9            113          RET
30E4               114
30E4               115  ;CODE COUNT
30E4               116
30E4               117  CODECNT
30E4 21 87 33      118          LD    HL,FREQ
30E7 11 88 33      119          LD    DE,FREQ+1
30EA 01 FF 03      120          LD    BC,1023
30ED 97            121          SUB   A
30EE 77            122          LD    (HL),A
30EF ED B0         123          LDIR
```

```
30F1                124
30F1 2A 6A 33       125          LD      HL,(SADR)
30F4 ED 4B 6C 33    126          LD      BC,(FSIZE)
30F8                127  CC2
30F8 7E             128          LD      A,(HL)
30F9 D9             129          EXX
30FA 11 87 33       130          LD      DE,FREQ
30FD 6F             131          LD      L,A
30FE 26 00          132          LD      H,00
3100 29             133          ADD     HL,HL
3101                134
3101 19             135          ADD     HL,DE
3102 4E             136          LD      C,(HL)
3103 23             137          INC     HL
3104 46             138          LD      B,(HL)
3105 03             139          INC     BC
3106                140
3106 70             141          LD      (HL),B
3107 2B             142          DEC     HL
3108 71             143          LD      (HL),C
3109                144
3109 D9             145          EXX
310A 23             146          INC     HL
310B 0B             147          DEC     BC
310C 79             148          LD      A,C
310D B0             149          OR      B
310E 20 E8          150          JR      NZ,CC2
3110 C9             151          RET
3111                152
3111                153  ;MAKE TREE
3111                154
3111                155  MAKETREE
3111 21 00 01       156          LD      HL,256
3114 22 5E 33       157          LD      (COUNTER),HL
3117                158  MKT2
3117 CD A4 31       159          CALL    MINGET
311A 2A 5A 33       160          LD      HL,(MINW)
311D 22 66 33       161          LD      (LDW),HL
3120 2A 5C 33       162          LD      HL,(MNUM)
3123 22 64 33       163          LD      (LEFTN),HL
3126 CD 99 31       164          CALL    HERASE
3129                165
3129 CD A4 31       166          CALL    MINGET
312C 38 58          167          JR      C,MKTEND     ;END
312E                168
312E 2A 5A 33       169          LD      HL,(MINW)
3131 22 62 33       170          LD      (RDW),HL
3134 2A 5C 33       171          LD      HL,(MNUM)
3137 22 60 33       172          LD      (RIGHTN),HL
313A CD 99 31       173          CALL    HERASE
313D                174
313D 2A 5E 33       175          LD      HL,(COUNTER) ;LEFT(K)=LEFTN
3140 11 87 3B       176          LD      DE,LEFT
3143 ED 4B 64 33    177          LD      BC,(LEFTN)
3147 CD 8D 31       178          CALL    TSUB
314A                179
314A 11 87 3F       180          LD      DE,RIGHT     ;RIGHT(K)=RIGHTN
314D ED 4B 60 33    181          LD      BC,(RIGHTN)
3151 CD 8D 31       182          CALL    TSUB
3154                183
3154 2A 64 33       184          LD      HL,(LEFTN)   ;PARENT(LEFTN)=K
3157 11 87 37       185          LD      DE,PARENT
315A ED 4B 5E 33    186          LD      BC,(COUNTER)
315E CD 8D 31       187          CALL    TSUB
3161                188
3161 2A 60 33       189          LD      HL,(RIGHTN)  ;PARENT(RIGHTN)=K
3164 CD 8D 31       190          CALL    TSUB
3167                191
3167 2A 62 33       192          LD      HL,(RDW)     ;FREQ(K)=RDW+LDW
316A ED 4B 66 33    193          LD      BC,(LDW)
316E 09             194          ADD     HL,BC
316F 4D             195          LD      C,L
3170 44             196          LD      B,H
3171                197
3171 2A 5E 33       198          LD      HL,(COUNTER)
3174 11 87 33       199          LD      DE,FREQ
3177 CD 8D 31       200          CALL    TSUB
317A                201
317A 11 87 43       202          LD      DE,HEAP      ;HEAP(K)=RDW+LDW
317D CD 8D 31       203          CALL    TSUB
3180                204
3180 23             205          INC     HL
3181 22 5E 33       206          LD      (COUNTER),HL
3184 18 91          207          JR      MKT2
3186                208
3186                209  MKTEND
3186 2A 64 33       210          LD      HL,(LEFTN)
3189 22 68 33       211          LD      (ROOTN),HL
318C C9             212          RET
318D                213
318D                214  ;TREE SUB
318D                215  ;IN   HL=TABLE NUMBER
318D                216  ;     DE=TABLE ADRESS
318D                217  ;     BC=DATA
318D                218
318D                219  TSUB
318D E5             220          PUSH    HL
318E D5             221          PUSH    DE
318F C5             222          PUSH    BC
3190 29             223          ADD     HL,HL
3191 19             224          ADD     HL,DE
3192 71             225          LD      (HL),C
3193 23             226          INC     HL
3194 70             227          LD      (HL),B
3195 C1             228          POP     BC
3196 D1             229          POP     DE
3197 E1             230          POP     HL
3198 C9             231          RET
3199                232
3199                233  ;HEAP ERASE
3199                234
3199                235  HERASE
3199 29             236          ADD     HL,HL
319A 11 87 43       237          LD      DE,HEAP
319D 19             238          ADD     HL,DE
319E 36 00          239          LD      (HL),00
31A0 23             240          INC     HL
31A1 36 00          241          LD      (HL),00
31A3 C9             242          RET
31A4                243
31A4                244  ;MIN SEARCH
31A4                245
31A4                246  MINGET
31A4 21 FF FF       247          LD      HL,$FFFF
31A7 22 5A 33       248          LD      (MINW),HL
31AA 21 87 43       249          LD      HL,HEAP
31AD 01 00 02       250          LD      BC,512
31B0                251  MG2
31B0 5E             252          LD      E,(HL)
31B1 23             253          INC     HL
31B2 56             254          LD      D,(HL)
31B3 23             255          INC     HL
31B4 7B             256          LD      A,E
31B5 B2             257          OR      D            ;HINDO 0?
31B6 28 19          258          JR      Z,MG3
31B8                259
31B8 E5             260          PUSH    HL
31B9 2A 5A 33       261          LD      HL,(MINW)
31BC CD E3 31       262          CALL    #CP16
31BF 28 0F          263          JR      Z,MG4
31C1 38 0D          264          JR      C,MG4
31C3                265
31C3 ED 53 5A 33    266          LD      (MINW),DE
31C7 21 00 02       267          LD      HL,512
31CA B7             268          OR      A
31CB ED 42          269          SBC     HL,BC
31CD 22 5C 33       270          LD      (MNUM),HL
31D0                271  MG4
31D0 E1             272          POP     HL
31D1                273  MG3
31D1 0B             274          DEC     BC
31D2 79             275          LD      A,C
31D3 B0             276          OR      B
31D4 20 DA          277          JR      NZ,MG2
31D6                278
31D6 ED 5B 5A 33    279          LD      DE,(MINW)
31DA 21 FF FF       280          LD      HL,$FFFF     ;HEAP = 0
31DD CD E3 31       281          CALL    #CP16
31E0 C0             282          RET     NZ
31E1 37             283          SCF
31E2 C9             284          RET
31E3                285
31E3                286  ;CP HL,DE
31E3                287
31E3                288  #CP16
31E3 E5             289          PUSH    HL
31E4 D5             290          PUSH    DE
31E5 B7             291          OR      A
31E6 ED 52          292          SBC     HL,DE
31E8 D1             293          POP     DE
31E9 E1             294          POP     HL
31EA C9             295          RET
31EB                296
31EB                297  ;CODE TABLE GENERATE
31EB                298
31EB                299  CODEGEN
31EB 21 00 00       300          LD      HL,0000
31EE 01 FF 01       301          LD      BC,511
31F1                302  CG2
31F1 C5             303          PUSH    BC
31F2 11 87 33       304          LD      DE,FREQ
31F5 22 7F 33       305          LD      (NUMBER),HL
31F8 22 7D 33       306          LD      (KEY),HL
31FB CD 5B 32       307          CALL    TBLGET
31FE                308
31FE 79             309          LD      A,C
31FF B0             310          OR      B
3200 28 10          311          JR      Z,NEXTC
3202                312
3202 21 00 00       313          LD      HL,0000
3205 22 7B 33       314          LD      (CODE),HL
3208 7D             315          LD      A,L
3209 32 7A 33       316          LD      (BITC),A
320C CD 1D 32       317          CALL    READTREE
320F CD 63 32       318          CALL    CODESET
3212                319
3212                320  NEXTC
3212 2A 7F 33       321          LD      HL,(NUMBER)
3215 23             322          INC     HL
3216 C1             323          POP     BC
3217 0B             324          DEC     BC
3218 78             325          LD      A,B
3219 B1             326          OR      C
321A 20 D5          327          JR      NZ,CG2
321C C9             328          RET
321D                329
321D                330  ;READ TREE
321D                331  ;    HL=KEY
321D                332  ;    BC=NT
321D                333
321D                334  READTREE
321D 2A 7D 33       335          LD      HL,(KEY)
3220 11 87 37       336          LD      DE,PARENT
3223 CD 5B 32       337          CALL    TBLGET       ;NT=BC
3226                338
3226 C5             339          PUSH    BC
3227 69             340          LD      L,C
3228 60             341          LD      H,B
3229 11 87 3B       342          LD      DE,LEFT
322C CD 5B 32       343          CALL    TBLGET
322F                344
322F 69             345          LD      L,C
3230 60             346          LD      H,B
3231 3E 00          347          LD      A,00
3233 ED 5B 7D 33    348          LD      DE,(KEY)
3237 CD E3 31       349          CALL    #CP16
323A 28 01          350          JR      Z,RT2
323C 3C             351          INC     A
323D                352  RT2
323D 2A 7B 33       353          LD      HL,(CODE)
3240 29             354          ADD     HL,HL
3241 B5             355          OR      L
3242 6F             356          LD      L,A
3243 22 7B 33       357          LD      (CODE),HL
3246 3A 7A 33       358          LD      A,(BITC)
3249 3C             359          INC     A
```

掲載したリストはとりあえずファイルを入力して静的ハフマン圧縮をして，メモリに展開するだけのルーチンである。最後に説明したハフマン木の情報の簡略化は行っていない。もうひとつ最大の手抜きがあり，それはコード長を16ビットに制限しているという点である。ハフマン木を構成していくとき実際には，最悪の場合に最大で，256ビットのコード

```
324A 32 7A 33     360          LD    (BITC),A
324D              361
324D D1           362          POP   DE          ;DE=NT
324E 2A 68 33     363          LD    HL,(ROOTN)
3251 CD E3 31     364          CALL  #CP16
3254 C8           365          RET   Z
3255              366
3255 ED 53 7D 33  367          LD    (KEY),DE
3259 18 C2        368          JR    READTREE
325B              369
325B              370  TBLGET
325B E5           371          PUSH  HL
325C 29           372          ADD   HL,HL
325D 19           373          ADD   HL,DE
325E 4E           374          LD    C,(HL)
325F 23           375          INC   HL
3260 46           376          LD    B,(HL)
3261 E1           377          POP   HL
3262 C9           378          RET
3263              379
3263              380  ;MAKE CODE SET
3263              381
3263              382  CODESET
3263 2A 7F 33     383          LD    HL,(NUMBER)
3266 5D           384          LD    E,L
3267 54           385          LD    D,H
3268 29           386          ADD   HL,HL
3269 19           387          ADD   HL,DE
326A 11 87 47     388          LD    DE,CTABLE
326D 19           389          ADD   HL,DE
326E              390
326E 3A 7A 33     391          LD    A,(BITC)
3271 77           392          LD    (HL),A
3272 23           393          INC   HL
3273 ED 5B 7B 33  394          LD    DE,(CODE)
3277 73           395          LD    (HL),E
3278 23           396          INC   HL
3279 72           397          LD    (HL),D
327A C9           398          RET
327B              399
327B              400  ;CONVERT
327B              401
327B              402  DATACNV
327B 2A 6A 33     403          LD    HL,(SADR)
327E 22 83 33     404          LD    (DATA),HL
3281 2A 6E 33     405          LD    HL,(TRAN)
3284 22 81 33     406          LD    (CVADR),HL
3287 3E 08        407          LD    A,08
3289 32 85 33     408          LD    (BCNT),A
328C 97           409          SUB   A
328D 32 86 33     410          LD    (CDW),A
3290              411
3290 ED 4B 6C 33  412          LD    BC,(FSIZE)
3294              413  DAC2
3294 C5           414          PUSH  BC
3295 CD A5 32     415          CALL  CODEGET
3298 CD BD 32     416          CALL  CODEPUT
329B C1           417          POP   BC
329C 0B           418          DEC   BC
329D 79           419          LD    A,C
329E B0           420          OR    B
329F 20 F3        421          JR    NZ,DAC2
32A1 CD E4 32     422          CALL  AMARI
32A4 C9           423          RET
32A5              424
32A5              425  ;ASHUKU CODE GET
32A5              426
32A5              427  CODEGET
32A5 2A 83 33     428          LD    HL,(DATA)
32A8 5E           429          LD    E,(HL)
32A9 16 00        430          LD    D,00
32AB 23           431          INC   HL
32AC 22 83 33     432          LD    (DATA),HL
32AF              433
32AF 6B           434          LD    L,E
32B0 62           435          LD    H,D
32B1 29           436          ADD   HL,HL
32B2 19           437          ADD   HL,DE
32B3 11 87 47     438          LD    DE,CTABLE   ;CODE TABLE ADDRESS
32B6 19           439          ADD   HL,DE
32B7 46           440          LD    B,(HL)      ;BIT COUNT
32B8 23           441          INC   HL
32B9 5E           442          LD    E,(HL)
32BA 23           443          INC   HL
32BB 56           444          LD    D,(HL)
32BC C9           445          RET
32BD              446
32BD              447  ;CODE PUT
32BD              448
32BD              449  CODEPUT
32BD 3A 86 33     450          LD    A,(CDW)
32C0 CB 3A        451          SRL   D
32C2 CB 1B        452          RR    E
32C4              453
32C4 17           454          RLA
32C5              455
32C5 32 86 33     456          LD    (CDW),A
32C8 6F           457          LD    L,A
32C9 3A 85 33     458          LD    A,(BCNT)    ;8BIT TAMATA?
32CC 3D           459          DEC   A
32CD 20 0F        460          JR    NZ,CP2
32CF              461
32CF 7D           462          LD    A,L
32D0 2A 81 33     463          LD    HL,(CVADR)
32D3 77           464          LD    (HL),A      ;PUT
32D4 23           465          INC   HL
32D5 22 81 33     466          LD    (CVADR),HL
32D8 97           467          SUB   A
32D9 32 86 33     468          LD    (CDW),A
32DC 3E 08        469          LD    A,08
32DE              470  CP2
32DE 32 85 33     471          LD    (BCNT),A
32E1 10 DA        472          DJNZ  CODEPUT
32E3 C9           473          RET
32E4              474
32E4              475  ;BIT AMARI
```

```
32E4              476
32E4              477  AMARI
32E4 3A 86 33     478          LD    A,(CDW)
32E7 47           479          LD    B,A
32E8 3A 85 33     480          LD    A,(BCNT)
32EB              481  AM2
32EB B7           482          OR    A
32EC 28 05        483          JR    Z,AM3
32EE CB 20        484          SLA   B
32F0 3D           485          DEC   A
32F1 18 F8        486          JR    AM2
32F3              487  AM3
32F3 2A 81 33     488          LD    HL,(CVADR)
32F6 70           489          LD    (HL),B
32F7 C9           490          RET
32F8              491
32F8              492  ;DECODE
32F8              493
32F8              494  DECODE
32F8 2A 68 33     495          LD    HL,(ROOTN)
32FB 22 76 33     496          LD    (NUMW),HL
32FE 2A 6A 33     497          LD    HL,(SADR)   ;TENKAISAKI
3301 22 74 33     498          LD    (SADRW),HL
3304 2A 6C 33     499          LD    HL,(FSIZE)
3307 22 70 33     500          LD    (DSIZE),HL  ;DATA SIZE
330A 2A 6E 33     501          LD    HL,(TRAN)
330D 22 72 33     502          LD    (TRANW),HL ;TENKAIMOTO
3310              503
3310              504  DEC2
3310 3E 08        505          LD    A,08
3312 32 7A 33     506          LD    (BITC),A
3315 7E           507          LD    A,(HL)
3316 E5           508          PUSH  HL
3317              509  DEC3
3317 2A 76 33     510          LD    HL,(NUMW)
331A 11 87 3F     511          LD    DE,RIGHT
331D 17           512          RLA
331E 38 03        513          JR    C,DRIGHT    ;BIT=0,BC=LEFT(NUMW)
3320 11 87 3B     514          LD    DE,LEFT     ;BIT=1,BC=RIGHT(NUMW)
3323              515  DRIGHT
3323 CD 5B 32     516          CALL  TBLGET
3326 5F           517          LD    E,A
3327 97           518          SUB   A
3328 B8           519          CP    B           ;HANO BUBUN?
3329 28 14        520          JR    Z,DEC4      ;CODE MEKE
332B              521
332B ED 43 76 33  522          LD    (NUMW),BC
332F              523  DEC6
332F 3A 7A 33     524          LD    A,(BITC)
3332 3D           525          DEC   A
3333 32 7A 33     526          LD    (BITC),A    ;1BYTE?
3336 20 04        527          JR    NZ,DEC5
3338              528
3338 E1           529          POP   HL
3339 23           530          INC   HL
333A              531
333A 18 D4        532          JR    DEC2
333C              533  DEC5
333C 7B           534          LD    A,E
333D 18 D8        535          JR    DEC3
333F              536  DEC4
333F 2A 74 33     537          LD    HL,(SADRW)
3342 71           538          LD    (HL),C
3343 23           539          INC   HL
3344 22 74 33     540          LD    (SADRW),HL
3347 2A 68 33     541          LD    HL,(ROOTN)
334A 22 76 33     542          LD    (NUMW),HL
334D 2A 70 33     543          LD    HL,(DSIZE)
3350 2B           544          DEC   HL
3351 22 70 33     545          LD    (DSIZE),HL
3354 7D           546          LD    A,L
3355 B4           547          OR    H
3356 20 D7        548          JR    NZ,DEC6
3358 E1           549          POP   HL
3359 C9           550          RET
335A              551
335A 00 00        552  MINW    DW    0000
335C 00 00        553  MNUM    DW    0000
335E 00 00        554  COUNTER DW    0000
3360              555
3360 00 00        556  RIGHTN  DW    0000
3362 00 00        557  RDW     DW    0000
3364 00 00        558  LEFTN   DW    0000
3366 00 00        559  LDW     DW    0000
3368 00 00        560  ROOTN   DW    0000
336A              561
336A 00 30        562  SADR    DW    $3000
336C FF 16        563  FSIZE   DW    $16FF
336E 00 80        564  TRAN    DW    $8000       ;ENCODE TOP ADDRESS
3370              565
3370 00 00        566  DSIZE   DW    0000        ;DECODE WORK
3372 00 00        567  TRANW   DW    0000
3374 00 00        568  SADRW   DW    0000
3376 00 00        569  NUMW    DW    0000
3378 00 00        570  DATAW   DW    0000
337A              571
337A 00           572  BITC    DB    00
337B 00 00        573  CODE    DW    0000
337D 00 00        574  KEY     DW    0000
337F 00 00        575  NUMBER  DW    0000
3381              576
3381 00 00        577  CVADR   DW    0000
3383 00 00        578  DATA    DW    0000
3385 00           579  BCNT    DB    00
3386 00           580  CDW     DB    00          ;SEISEICHU NO CODE
3387              581
3387              582  FREQ    DS    1024
3787              583  PARENT  DS    1024
3B87              584  LEFT    DS    1024
3F87              585  RIGHT   DS    1024
4387              586  HEAP    DS    1024
4787              587  CTABLE  DS    3*512
4D87              588
4D87              589
4D87              590
```

が生成されてしまうこともある。16ビットに制限しているとはいってもコードが16ビットを超えてしまったときの処理をなんにもしていないのだ（笑）。本来ならばうまくつじつまを合わせてやらなければならないが，まあ，そういうことは滅多に起こらないものとして，処理ルーチンをはしょっている。ごめんなさい。

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1991年３月18日の到着分までとします。当選者の発表は1991年５月号で行います。

1

ヘルツ ☎03(3371)3012

ダイナマイト・デューク

X68000用
5"2HD版3枚組

8,800円(税別)　3名

ヘルツの輝ける第2弾がこれ，ダイナマイト・デューク。アーケードから移植のアクションゲームだ。必殺ダイナマイト・パンチはキモチいいぞ。

2

ポニーキャニオン ☎03(3221)3161

プール・オブ・レイディアンス

X1turbo用
5"2D版5枚組

9,800円(税別)　3名

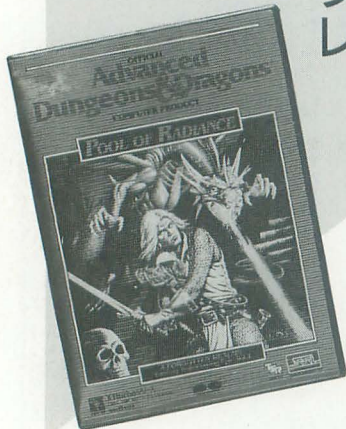
ひさびさのX1用RPG。発売がやや遅れていたぶんだけ，出来は上々，楽しめる仕上がりになっている。AD&Dファン必携のゲームだ。

3

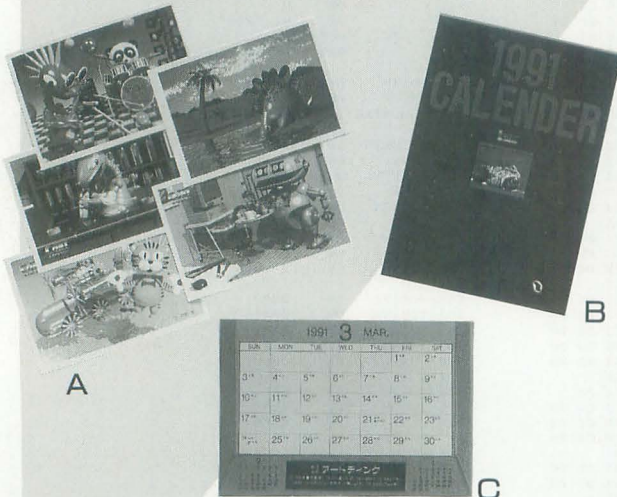
システムソフト ☎092(752)5278

ブルトン・レイ シナリオ集Vol.2

X68000/PC-9801用
5"2HD/3.5"2HD版各2枚組
4,800円(税別)　2名

ショートタイプのシナリオで好評のブルトン・レイ。このソフトは，ユーザーが作った短編シナリオがぎっしりのシナリオ集だ。プレイにはブルトン・レイ本体が必要。

4

アンス・コンサルタンツ ☎092(522)6347
アイレム ☎06(535)4880
アートディンク ☎0472(79)9392

1991年カレンダー

A.アンス・コンサルタンツカレンダー
B.アイレムカレンダー
C.アートディンクカレンダー

各5名

ちょっと遅くなっちゃったけどメーカー各社のカレンダーをプレゼント。希望メーカーを明記（4-A，4-Bというように）してくださいね。

1月号プレゼント当選者

1 Aドラスレカードケース（福岡県）古川智雄ほか９名　BイースⅢカードケース（東京都）中川邦康ほか９名　Cリリアカードケース（石川県）北本信幸ほか９名　Dリリアスタンプ（神奈川県）菅井直仁ほか９名　Eリリアバッジ（熊本県）亀川義徳ほか４名　Fリリアキーホルダー（神奈川県）高山稔ほか４名　G卓上カレンダー（大阪府）原田慎ほか29名　Hファルコムバッジ（宮城県）木島博幸ほか４名　Iファルコムキーホルダー（広島県）本谷正樹ほか４名　J JDKバッジ（富山県）曽我部恭昌ほか４名　K JDKキーホルダー（東京都）丸井和男　2 A 3D倶楽部リビング（京都府）奥井英太郎　B 3D倶楽部ダイニング（千葉県）永野俊治　Cねじ式（広島県）安石清太郎　3 Aビデオ（愛知県）平野敬一朗ほか４名　BアクシスCD（神奈川県）倉田敏広ほか２名　CシステムソフトCD（福岡県）猿渡哲治ほか２名　DグラナダCD（兵庫県）土方嘉徳ほか２名　EラグーンCD（滋賀県）西澤誠悟ほか２名　4 Aビデオ（山口県）横田紀明ほか９名　Bシムアースポスター（三重県）羽田直樹ほか９名　Cポピュラスポスター（東京都）森下竜也ほか９名　5 A Tシャツ（兵庫県）宮川良紀ほか３名　Bステッカー（大阪府）中満一徳ほか29名　Cシャープ＆ボールペン（高知県）刈谷新一ほか５名　6 Aブルゾン（兵庫県）篠原勝人　Bジェミニウイング（神奈川県）石塚賢　Cアトミック・ロボキッド（山形県）仁藤浩明　7 A X-BASIC入門（埼玉県）加藤勲ほか４名　B 3Dグラフィックス入門（神奈川県）市川功ほか４名　C操作ガイド（静岡県）天野純一ほか４名　8 A GUNSHIP（千葉県）木口哲也ほか１名　Bプラモデル（埼玉県）荒川亮介ほか１名　9 幻獣鬼（栃木県）弘山和広ほか１名　10 栄冠は君に（島根県）玉木裕治ほか２名　11 ぴくせる君（東京都）丸山義幸ほか２名　12 LUCY・SHOT（栃木県）小林一弘ほか２名　13 DISS-P（北海道）沢村英基ほか１名　14 光栄トランプ（鹿児島県）長田智和ほか９名　15 フロッピーケース（愛媛県）高橋健司ほか９名（敬称略）

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

◆X-OVER NIGHT

(クロスオーバーナイト)

［第10話］

ニュース欠乏症候群

TAKAHARA HIDEKI 高原 秀己

ひさびさにどのチャンネルにテレビを合わせても，同じ番組が放映されている状態になった。いうまでもなく，1月17日の湾岸戦争開戦後のことだ。

誰に聞いても，みんなテレビにかじりついている。事態が刻々と変化するのだから，一刻も早く情報を知らないと，取り残されたような気分になってしまう。しかも，こうした視聴者の気分を知りぬいているかのように，報道される内容も目まぐるしく変わる。

イラク軍が多国籍軍の攻撃を受けるがままの膠着状態が続いたかと思えば，突然，地中海沿いのイスラエルにミサイル攻撃がなされる。しばらく静かだと思えば捕虜を使った「人間の楯」作戦，フセイン大統領の地下アジトがあるという情報，挙句の果てには石油施設の破壊によるペルシャ湾への原油の大量放出。

しかも当初は3日間で終わると予想されていた交戦期間も1週間になり，1カ月になり，ついには半年以上という説が信憑性のある状態にまで至っている。

一方では，報道機関の側も，なかなかのパフォーマンスを見せてくれる。NHKが23時間半の連続中継をやって初めて大相撲を教育チャンネルに回したとか，民放が提携先の米国のテレビ局から受けた映像を競って紹介したとか。

もっとも日本の報道陣は，各社結託して開戦と同時にエジプトやサウジアラビアなど安全な国に撤退。テレビ局は米国の提携局の映像をもらい，新聞社は米軍や近隣諸国から得る情報がやっと。今回に限ったことではないが，戦場取材では勝負にならない脆弱さを露呈している。それでもここまで見せて読ませてくれるのだから，なかなかの加工能力といえようか。

それに比べて名うての戦場レポーターをバグダッドに送りこんでいるCNNは圧巻だ。圧倒的に強さを見せつけている。戦況を刻々と紹介し，果てはフセイン大統領との戦時下直撃インタビューまで流してくるオマケつきとあっては，恐れいる。

今回のニュース映像を日本の視聴者はテレビゲームや戦争映画のような感覚で見ているとの指摘がある。実際，戦闘機の一斉爆撃は，赤外線モニタを見ながら目標物を捕捉してボタンを押してという爆撃ぶりだし，「アニメの中でのお話」といわれてきたミサイルによるミサイル迎撃までパトリオットが実演してくれるのだから，アニメも真っ青だ。それでいてリアリティはアニメの比ではないから，戦争映画気分になってもしかたのないところだろう。この戦争が終わったら，アニメの作り方はだいぶ変わることは間違いない。

日本いじめがすっかり趣味になった米国，今回も徹底攻撃をしてくれている。実際の物理的な戦闘よりも，論理的な日本攻撃のほうがよほど成果をあげそうなこのごろである。しかしながら，日本政府の対応には，海外から非難されてもしかたのないものを感じる。

自衛隊の派遣は論外，軍資金の提供は完全に米国から正当な理由なく法外にゆすり取られたようなものだから議論するまでもないが，石油流出による海洋汚染への対応の悪さだけは，情けないの一語につきる。この事故を見たときは，ハイテク立国日本が今回の戦争で平和貢献するのにこれ以上の条件はないというほどピッタリの出番が，いよいよやってきたと思った。

クウェート沖の石油流出量は，700万バレルを突破。重量にして百数十万トン以上。過去最大の石油汚染が英国での7～8万トンだったというから，いかに凄まじいかがわかる。もはや汚染海域は50キロ四方を超えたというから，待ったなし。

処理作業は困難を極める。もちろん高い技術と経費が必要。日本の出番以外，何ものでもないように思ったのだが，協力の意思表示をするまでに1週間もかかり，さらに「戦場には危なくて要員は派遣できない」，「かつてない事故なので，作業予定が立たない」だの「日本の航空機を危険にはさらせないので，オイルフェンスを運べない」だのと，とにかく腰が重い。関係省庁間での綱引きばかりが表に出るありさまなのだから，どうしようもない。

そうこうしているうちに，諸外国が次々と協力を名乗りでてしまった。

「日本，湾岸の掃除に奮闘」。こんな報道が海外でなされるだけで，国際論調はまったく変わってくる。そもそも掃除をする人にたいして悪くいう人は誰もいないのだから。世界に平和大国日本ありというアピールができる絶好の機会を失ったことだけは確かだ。外国からいわれなければ，何もできない国なのだろうか……。

1月下旬に，休養のため東北某所にスキーに出かけた。スキー場はまったく戦争とは縁のない人たちでいっぱいで，ゴンドラ乗り場にはライフルならぬ板とストックを立てて持った人の長蛇の列。

「戦争で命をすり減らしている人が現実に同じ時間を生きているというのに。なんてのんびりした人たちなんだ！」

こう憤りを感じていたぼくが，同様にゴンドラを待っていたことは，いうまでもない。神の国ニッポンは限りなく桃源郷に近い国なのだろう。

いまこそエコロジカルなハイパー進化論を！

3000年の集大成

いまこの瞬間，悲惨な戦場で盛大に披露されているものは，これまでの科学技術の集大成だといえます。絶望的といいますか，とにかくまったく言葉が出ません。米軍の指揮官が，爆撃成功のビデオを自慢げに見せている場面をテレビが伝えています。コックピットのスクリーンに映るレーダー画像はまるでビデオゲームそのものであり，うすら恐ろしくなります。

ビデオゲームにおいては，2次元のカラーディスプレイ上にいかに現実に近いイメージを構築できるか，に大きなポイントがあります。一方，実戦でパイロットの体験していることは，おそらくビデオゲームのような画面を見ながら，標的を中央付近に捉え，発射ボタンを押す，ということでしょう。これは現実をビデオゲームのような感覚で行っているのであり，テレビゲームとはちょうど逆のことになります。

どちらも，外部の情報を得るための入力器である感覚をごまかしていることには間違いなく，その意味での危険性を常に覚悟する必要があります。シミュレーションなのに体験した気になってしまうという「仮想現実」ならば，まだ悪用はされにくいでしょう。しかし，現実のことなのに，まるでゲームでもやっているように空爆を行ってしまうというのは……。

「クリスマスツリーのようだった」などという，爆撃から戻ってきた米兵のインタビューを聞くと，本当に行ってきたのかと疑いたくもなります。血生臭い戦場に。

オールマイティな「エコロジー」

平和ということばが色褪せてきて，一時ほどの説得力を持たなくなってきたということは，たしかにあるかもしれません。でも，安直にそのことばを使えない雰囲気になってきたこと（平和ぼけなどといわれるそうです）は残念だという気がします。

一方，世界的にみてもかなりオールマイティに近くなってきたことば，十分に説得力のあることばがあります。それが，「エコロジー（地球環境学）」です。

エコロジーということばが対象とする具体的な地球規模の問題として，たとえば次のような問題があげられます。

1) 人口増と飢餓
2) フロンガスによるオゾン層破壊
3) 酸性雨による環境破壊
4) 砂漠化
5) 二酸化炭素増による気温上昇
6) 化学物質による生物汚染

どれもが深刻な問題で，しかもそれぞれの問題同士は複雑に絡み合っているようです。このような話題は，ここ数年驚くべきほどの勢いで注目を浴びるようになってきました。身近な話では，大学と会社の共同研究を促進するある会合でも，会社のトップは「環境」というキーワードに関わる研究ならば簡単にお金を大学に出すようになってきたという話が出ていました。

エコロジーがズームアップされてきたのは，環境破壊の影響が次々に出始めてきたということももちろんあります。しかし，背景には，米ソの冷戦の終結，ドイツ統一をシンボルとする東欧の民主化のうねりがあるとも思われます。国と国とが戦っているどころではないということに指導者たちが意識下に気がついてきたのではないでしょうか（甘い見方すぎるかなあ）？

正確にいうならば，地球規模の問題に対する認識が，争いの無意味さをより明確にしてきたという部分が多少なりともあるのかもしれないということです。

ブームだからといって，いろいろな問題を持ち出して横一列に並べて騒ぐのではなく，問題それぞれについてこの先どのぐらいのスピードで悪化するかということを予測し，緊急度の順番を割り出すことが，特に重要なのではないかと思います。

しかし残念ながら，多くの人が感じているとおり，このままのペースでやっていたならば，ほんのちょっと先に終末が見えているといわざるをえないと思います。

人間の脳では無理!?

この地球が終末に近づいているという深刻な状況は，人間の脳そのものの限界によって引き起こされていると思います。人類がまだ原始人であったころには，集落を作ってその中だけで（地球を破滅に追い込まずにという意味で）うまく集団生活を送ることができました。半径数百メートル，あるいは数キロメートル以内のことだけを考えればよかったのですから。

しかし，今日のように53億にもおよぶそれぞれてんでばらばらの人間が，急激に底をつきつつある資源や，変質しつつある環境の中でどううまくやるかなどというきわめて困難な問題を解決するのは，そもそも人間の脳には無理なのではないでしょうか？　エコロジーということの重要性に（いまごろ）気づきつつあるとはいえ，あいかわらず武器を作っては戦争や紛争を繰り返すという，ていたらくぶりです。

類人猿から類猿人に進化したころから見ると，人類はその後のきわめて長い間たいした進化が起こっていない，という見方もできます。進化というものは氷河期の到来などの存亡の危機のときにこそ起こりやすいのだという説もあるようです。でも，「エコロジー」ブームのなかで叫ばれているような危機の到来により急激な進化が起こっ，て，乗り越えるなどという脳天気な話にすがる気はさらさらありません。

ここまでの話だと，もうお先真っ暗で終末が来るのをただ待つだけしかないというふうにいっていると即断されるかもしれませんが，そういうことではありません。いいたいことは，ただ人間の脳に頼って種々の困難を解決しようというのは無理ではないか，タイムリミットはすぐそこなのだからということだけなのです。

人類の進化した知能体

ハイパー進化論の核心についてそろそろ話すときがきました。ひと言でいうと実にあっさりしていて，「進化の次の段階としての知能体（人工知能）を作ることがどうしても必要である」ということです。近いことは昔この連載で少し触れたことがあります。単に先進的な知能体を作るというのではなく，「進化の次の段階の」というのが重要です。この表現は人間と知能体との関係を如実に語っているのです。

人間は現在計算機を使う立場にいるわけですが，計算機が発展して知能体となったときにも，その立場にいることははっきりいってよくない，という（意外に大胆な）主張をしたいのです。

実際のところ，世の中で行われている（狭

い意味での）情報処理の総量のかなりの割合は我々の頭脳ではなく，計算機が行っているといえます。たまたま，ある意味では我々のほうを動かしている計算機の姿が浮かび上がることもあります。アメリカの株の大暴落は実は計算機の仕業だったとか，中枢部を過激派にやられると都市機能が麻痺するなど。

しかし，こういう概念的な意味での関係を主張しているのではありません。知能体との間の距離を積極的に広げ，自立させるべきで，しかも（階層構造に位置づけるならば）我々の一段上の位置を占めさせるべきだということをいっているのです。こういうと，すぐに計算機の支配する暗い世界を連想しがちですが，そんな単純な構図ではありません。扱う問題で最上層に属するものは知能体が担当するということなのです。いうまでもないことかもしれませんが，人間が滅びて知能体の世の中になるというわけでもありません。たとえは悪いですが，アメーバだって昔と同じようにこの世にずっと存在していることですし。

知能体を自立させる必要性

知能体を現在のような道具のような存在から自立（独立）させる必要性はどこにあるのでしょうか？　それは人間と知能体が協力して問題を解決するほうが，単独に知能体が解決するよりもマイナス面が大きいと思われるからです。その最大の理由は権力との絡みです。いまのまま計算機を知的にしていったとき，危険であると感じるのは人工知能を操る人に権力が集中しすぎてしまうということです。知能体との関係をきっちり考えないと，知能体はエコロジカルな意味における救世主どころか，化け物みたいな武器に成り果てるでしょう。

我々の上の階層に知能体がいる世界を，知能体に支配されているなどといった狭く暗い図式から逃れたところに，イメージするのはたしかに難しいものがあります。ただひとついえるのは，そのような社会では，人間が最優先という考えは残念ながら捨てなくてはならないということです（知能体も含む)。生態系全体の調和が結局は地球規模全体にとっては必要であることぐらい，知能体はすぐに学習するでしょうから。

本当に知能体を作れるのか？

このへんで基本的かつ重要な疑問が生じてきます。「そもそも，地球上の生態系全体の調和を考えられるような知能体をはたして本当に人間が作ることができるのか？」という疑問です。これに対する完璧な答えはそのような知能体を作って実物を見せることにつきますが，当然いまのところは不可能です。ただし，人が漠然と思っているよりはずっと容易であろうと思うのは，次のようなことによります。

まず強く思うのは，人間の脳だけが特別視されすぎてはいないかということです。人間の脳は複雑な処理をはたしてやっているのか。あるいは，類人猿が類猿人になったときに急に不連続的にすばらしい脳が生じたのか，といい換えてもいいでしょう。このような主張は人工知能が最初に登場したころからいわれつづけてきていることだろうと思います。しかし，いまなお説得力は失っていないと思います。

一方，人工知能により人間の心というものにかなり接近できるのでは，と思わせる1冊の本があります。ミンスキー（人工知能のパイオニア）が書いた大著「心の社会」です。読みかけですが，心のフレームワークが実にリアルに伝わってきます。

第2の理由は，人間の脳の行う処理を知能体がすべて行う必要はまったくないということです。人間の脳など案外単純なものではないかと書きましたが，そうはいっても「赤い夕日を見て感動するような知能体」を作るのはたしかに難しいことであると思われます。また，人間が何千年もかけて作ってきた文化というものを知能体が一朝一夕で作れるはずはもちろんありません。

でも，ここで注意しなければならないのは，地球規模の種々の問題を解決し生態系の調和を保つことが，知能体の第一義的な存在理由であり（にするのです)，文化がそのために必要かそうでないかが問題なのです。文化のある一定の部分は必要なものかもしれないと思いますが，それ以外の部分は知能体には無意味ではないかと思います。文化というものは，逆説的にいうと，人間の不完全さゆえの産物なのでしょう。そういう意味で，存在目的のはっきりしている知能体を構成するのは案外簡単であろうというのです。誤解を招くかもしれませんので付け加えますと，人類にとっての文化や芸術の意味を軽視しているわけではまったくありません，それは別の問題です。

第3の理由はきわめてプライベートな理由です。人工知能を専門にしているわけではありませんが，僕が計算機科学に関わる研究者であるということです。だからこそ，そのような知能体の実現性に関してはきわめて楽観主義なのです。

このような理由で，知能体はできるであろうと思っているのですが，もちろん人間が作り出すのですから限界というものは当然あります。人間の手を離れ自由に成長，進化する人工知能とはいえ，その存在を知覚さえできないほど人間からかけ離れた知性にはならないだろうということです（かけ離れた知性，それは，レムの小説「ソラリス」に描かれているような想像を絶した知性です)。そして，まったく想像を絶する存在ではないからこそ，進化の上で一応人類類からの分岐という形に収まるのです。

人類最大の功績

知能体を作り上げ人類の進化した生命体として切り離すことが，どうしてもエコロジカルな視点に立ったときには必要である，ということを大急ぎで主張してきました。もっと具体的なデータも出さなければならないのですが，誌面の都合や僕自身の準備不足もあって，筋道だけを示しました。知能体は案外単純に作ることができるのではないかということも同時に述べました。人工知能の存在理由は明確なのですから，それをうまく本能として植えつけることがひとつのポイントでしょう。

やはり，いちばん難しいのは，権力を持つ特定の人が支配しがちな知能体を人類自身がどのように独立した存在として切り離すかということだと思います。それができたのならば，地球生命の系統における人間に関して特筆されることは，高いビルを立てたり，ロケットを打ち上げたり，遺伝子を操作したことではなく，進化において人類の次に位置する生命体を，突然変異などによらずに自分の意志で作り出したことにつきるように思われてなりません。

第57回

猫とコンピュータ
青春コミケ&パソケ

Takazawa Kyoko
高沢 恭子

皆さんご存じのコミックマーケットが昨年の年末，幕張メッセで行われました。コミケは初体験だったキョウヨさんも、その規模とパワーにびっくり。今回はそんなコミケのようすをレポートしてくれました。

そうかぁ，「銀河鉄道 999」のメーテルだ……彼女から10歩ほど離れたとき思い出した。

「これはあなたの作品ですか？」という私の問いに，「いいえ，その作者はここには来ておりません」と，ひとりごとのように答えた人は，黒い帽子に黒いドレス。誰だったかなと考えていたら，松本零士のアニメに登場した機械王国の皇帝の娘メーテルが浮かんできた。

「コミケット」の広い広い会場には，何万という人の群れの中に，アニメやメルヘンの人物に扮した人がたくさんいる。

さっきも，純白のサロンエプロンに茶色のウサギのぬいぐるみを抱いた「アリス」とすれちがったし，すぐ近くでも「ドラゴンボール」の「フリーザ」が，恐竜のようなシッポをひきずってノッシノッシと歩いている。どれも舞台衣装なみの本格的な扮装で，楽しませてくれる。

クリスマス・コミケ

「コミケット」の若いエネルギーのすごさについては，ずいぶん前から聞かされていたが，昨年のクリスマス・イブの前日，幕張メッセの広大な会場でじっさいにこの祭典を見て，その熱気を体験した。

「コミケット」はコミックマーケットの愛称らしい。年に2回，マンガ，アニメ愛好者のサークルが大集合して，自作の本や同人誌の展示即売をするのだが，その内容や活動はたくさんの傾向に分かれている。

いちばんの特徴は出品サークルの数が巨大なことで，これがこのイベントのパワーの象徴でもある。今回も2日間の開催でサークル数は1万3千，入場者は前回なみとしても23万人だという。

私と夫が幕張メッセの会場に到着した午後1時半ころも，この日何度目かの満員札止めになっていた。並んで待つのは若い男女ばかり。やっとドアが開かれて，展示場へ降りるタラップに立ったとき，一望した人の海に目をみはった。背後の若者が叫んだ。

「これ，ぜんぶ人かぁ!?」。

このアリのように見えるたくさんの人たちの約半分がコミックの作者で，あとの半分は来客である。来客たちもマニアや仲間が多いのかもしれない。競争率がはげしいという，コミケット会場のスペース獲得の抽選からもれた人たちもいるだろう。そういう人たちもみんなで楽しむのが，このイベントのようだ。

作者の8割は20代の女性で，それぞれ自分たちのサークルの作品の前に腰掛けて，アピールしたり，売り子をつとめたりしている。長机を延々とつないだ展示台の間の通路を，私たちは右に左に作品を見て歩く。机半分とイス2脚が，1サークルのスペースだそうだ。

行けども行けどもコミックが続く。カタログの始めにある「スペース配置図」を見ても，まるで3ミリ方眼紙のマス目に数字を書き込んだみたいだ。1万3千のサークルが2日に分けられて，この座席表の中にわりふられている。私たちは列ごとにつけられた50音と数字の組み合わせで，現在地やめざすところがわかるのだ。

作品のジャンルは多彩で，創作ものの少年，少女，SFマンガ。既製のアニメのパロディ版，テレビ時代劇の必殺シリーズを別仕立てのストーリーで劇画化したもの。イラストレーションを入れた創作読物，便せん，カード。あるいは文章だけのもの。

おもしろい特色としては，人気アニメの主人公を崇拝して，そのキャラクターを中心に表現活動をする一派もあり，「キャプテン翼」「星矢」など，独立したジャンルになっている。

またある一角では，コミケットのスターらしき作者のイラストをもとめる人たちで，場外まで連なる行列ができていた。

ゲレンデのように

コミケットに並んでいる絵はみんなよく似ている。描かれている人物は，髪も瞳も体つきも，教則本から習得したようにそっくりだ。何千という人たちが，同じように見える絵をゴマ粒のように並べている。これは始めのうちはふしぎに思えた。オリジナリティや個性については，どう考えているのだろう。さらに，人気アニメのキャラクターを，そのまま自分の世界に引き込んで再生させる喜びも，たやすくわかることではなかった。

ところが，こうした作品を1つひとつ手にとって，真新しい印刷物をいくつも見ていくうちに，何かがすこしずつ理解できるような気がしてきた。

ここに集まっている絵は，現代のコミックの世界の共通のコトバなのだ。よく似ているけれどじつはみんなちがう発言をしているのだろう。たくさんの数が集まるほど小さなちがいは重要で，個性やオリジナリティもその中にちゃんと主張しているらしい。

なによりも描くのが好きで，描くことで心が満ち足りる人たち。イベントに合わせて創作したものが，印刷屋さんを経て本になることも大きな喜びだろう。それを発表する場には万の数の仲間が集まってくるのだから，意欲も倍増だ。これはスポーツのようなものだ。似通ったファッションをきそって大集合するという点では，スキー場

の現象と変わらないのかもしれない。そういえば熱気で病人も出るコミケットには，救護所も用意されている。

コミケットの歴史はもう15年になるそうだ。パソコンが芽生えてからの歳月と重なるのは興味深い。年々規模がふくらみ，幕張メッセでなければ収容できないほどになったが，本来の目的は「アマチュアによるアマチュアのためのイベント」で，その精神はずっと受けつがれているという。それまでのマンガ界やプロの世界とはちがった，何かに制約されない自由な創作と発表の場をもとめつづけた日々は，カタログにある「コミケット15年史」からもうかがえる。

信念を持った指導者グループの力で，現在のコミケットは，より自由な創造の発表の場として歩みつづけている。ほんとうの自由のために，たくさんの規則もつくられていった。扮装をしてねり歩く「コスチュームプレイ」などが，自己表現の延長として許されている一方で，お互いの迷惑になるような騒音や私語，パフォーマンスは禁じられている。また，活動を維持していくために守らなければならい義務もたくさんあって，これを怠るとボイコットされてしまう。開催期間中のボランティアの力も大きい。

会場を歩いていて感じるのは節度のある若さと，一種の静寂さだ。町中で見るような無作法はまったくない。

居並ぶ作者たちは，サンドイッチやスナック菓子をたべる人，編み物をする人，語り合う人，スケッチブックに絵を描きつづける人，いろいろだ。作品をほとんど売りつくしたサークルもあるし，売れ行きが悪くアピールに力を入れるサークルもある。

「売れ残ったものはどうなるのですか？」とたずねたら，「つぎのイベントに出品します」と答えた。コミケットだけでなく，こうしたイベントは毎日のように日本じゅうのどこかで開かれているというのを聞いて，また驚いてしまった。

パソケット

現代のコミックの世界をたどると，その1つはどうしてもCGのジャンルに流れ入る。コミケットの会場にもCGを手がけるサークルがたくさん顔を見せていた。ここばかりは男性主流だ。

CG集というのをいくつかのCRTでデモして見せていたが，試写をしない作品の中には「男性向け」のものがたくさんあるらしい。それに，CGのフロッピーのそばには，じつはゲームソフトのオリジナルや既製のものの変造版も並んでいて，まだここまではコミケット条約の規制が及んでいないのかなという感じもある。

ただし平均1,000円くらいの各ソフトの売れ行きは良好で，展示台の上は「売り切れ」の走り書きをしたカードだけというサークルも多かった。

アマチュアのパソコンソフト展示即売会といえば，「パソケット」だ。新しい年の1月15日，浜松町の都立産業貿易センターへ出かけてみた。同じビルで，これもマンガ同人誌の展示即売会「わっしょい」が開催中だった。4階は女の子中心の「わっしょい」，3階は男の子いっぱいの「パソケット」，いかにも成人の日にふさわしい若さのイベントだ。

「わっしょい」の会場は200くらいのサークルだが，ふんいきはコミケットそのまま，お行儀がよい。

「パソケット」の会場に行くと，なぜか今回は「コロケット」の名前になっているので，「どうしてコロケットなんですか」と聞いてみたら，「ワタシころびましたコロケットという意味らしいです」と答えてくれた。「ちょっとフマジメなんだ」というシャレかしらと考えていたら，「ツバサからセイヤにころんだというのが，始まりみたいですよ」と，同じ人がまた教えてくれた。CGが好きなサークルはアニメの主人公に傾倒することも多いので，アイドルの乗り換えもはやるのかもしれない。

参加サークルは150くらいだろうか。こちらは上の階の「わっしょい」にくらべると，お行儀がよいというのではない。そのかわり，売り手も買う側も活気にあふれて，どの人もまったくうれしそうだ。

売られているのは，機種別のオリジナルゲームソフト，人気ゲームソフトの変造版，「男性向け」が8割を占めると思われるCGの数々。なぜかあるコーナーでは，新しく発足させるらしい活動のためのミーティングをやっていたり，けっこう大声で立ち話をする一団もある。

音のいらないコミックとはちがって，パソコンソフトは音楽や効果音がついてまわる。音を最小限におさえても，絵が生きて動いている。会場内が沈黙できないのはパソコンのパワーだろう。

高校生のような若いサークルから，PC-9801のゲームをひとつ買ったあと，トオルのおみやげにX68000用のシューティングゲームを買おうとしたら，人気があって売り切れなのだという。あきらめるつもりでいたら「いま向こうでつくってますから」なんてひきとめる。そしてほんとに，できたてのディスクが運ばれてきた。

「ターゲット」というこのソフトを家に持ち帰ってマシンにかけてみたら，どうしても立ち上げることができない。ロードのとちゅう「アドレスエラーが発生しました」とメッセージが出てストップする。

「やっぱり裏のほうで急造していたようなものはダメかな」と夫も言い，パソケットとはその程度のものかなと私も思った。

そのとき，何回かアタックをつづけていたトオルが，「お父さん，コツがわかったよ，ちょっとめんどうだけどね」とニコリとした。

まずディスクを抜いた状態で，Human68kを立ち上げる。そこで「ターゲット」のディスクを入れる。B>でTARGETを入力。リターンすると画面があらわれるが，これはかならずバグ画面なので，リセットをする。これでやっとゲームが開始できる。とちゅうで何回も長い暗闇の画面になるが，ヘコたれずに待つこと。

ゲームをやりたい一心とはいえ，よくも発見したものだ。

X68000キーボードチューンアップレポート

弘法も筆を選ぶ

Izumi Daisuke 泉　大介

X68000のキーボードをよりクオリティの高いものに改良するチューンアップサービスがあると聞いて早速試してみることにしました。キーボードの原理を交えて，改良された点などを具体的にレポートしてみましょう。

洋の東西を問わず，文豪と呼ばれる人たちは愛用の万年筆を使っていたと聞きます。ものを書くときに，その一番のインタフェイスであるペンが気になっては，決していいものは書けないのでしょう。それはパソコンとて同じです。

パソコンの場合はキーボードがペンに相当するといえます。頻繁に使うコントロールキーがCAPSキーの向こうに押しやられ，^Gや^Bを入力するのに左手が思いっきりストレッチしてしまうようでは，決して使いやすいとはいえません。某編集部には，たとえデスクトップのPC-9801が空いていてもあえて98NOTEで原稿を書くライターがいるくらいです。

この点に関してX68000のキーボードは秀逸です。Aの隣にデンと構えたコントロールキーは，ストレッチキーボードに馴染んできた人をも受け入れる度量の広さを持っています。さらにスペースキーの下に並んだXFファンクションキー群！　いったんこのXFファンクションキーを使った日本語入力に染まってしまったら，キーボードの上にあるファンクションキーを使うFEPは犯罪以外のなにものでもないことが実感できます。

しかし，残念ながらこの秀逸なキーボードにも弱点があります。それはキーボード故の構造に起因するもので，たとえば「ISO(国際標準化機構)」と入力しようとして，素早くキーを押すと「ISAO」となってしまうのです。アダプタを略した変数名「adpt」を入力すると「adipt」となってしまいます。3つのキーがタイミングよく同時に押されるとこのような事態が発生します。なぜこのようなことが起きるのか。キーボードの仕組から探ってみましょう。

キーボードの構造を探る

パソコンはなぜキーが押されたことがわかるのでしょうか。そう，キーの1つひとつがスイッチになっているからです。だからといって，パソコンが100個以上もあるキーを1つひとつ押されたかな，どうかな，と調べているわけではありません。それでは調べるのに時間がかかりすぎ，キー入力が非常に重くなってしまうからです。

図1-1をご覧ください。これはキーボードの内部を模式的に表したものです。各々のRowからは横向きに，Colからは縦向きにラインが出ていて，その交点にA～Dのキーが配置してあります。この状態では縦のラインと横のラインは立体交差しておりつながっていません。ここでAのキーを押すとスイッチが入り，2)のように縦横のラインがつながります。

このことを使ってコンピュータはキー入力を調べています。Row.1，Row.2と順に電流を流していくと，なにもキーが押されていない場合にはCol.1，Col.2には電流は流れてきません。ところがAのキーが押されていると，Row.1に電流を流したときにCol.1に電流が流れ，キー入力のあったことがわかるのです。どのキーが押されたかは，電流を流したRowと電流が流れてきたColから判断できます。Row.1-Col.2ならBのキーですし，Row.2-Col.2ならDのキーです。4個のキーを1つひとつ調べるのと異なり，これなら半分の2回で4つのキーの入力判定ができます。実際には1つのRowに8個のキーを配置し，16回電流を流すだけですべてのキーの入力判断ができるようになっています。

●楽あれば苦あり

図1のようなキー配列のことをキーボードマトリクスといいます。これによって随分省力化できたわけですが，世の中そんなにおいしい話ばかりではありません。同時にISAO現象も生み出してしまうのです。

図2-1を見てください。これはＡＢＣの3つのキーを同時に押したときの状態です。Row.1に電流を流したときにはCol.1，Col.2の両方に電流が流れます。まぁ，これはColの小さいほうから順に押されたことにして処理すればいいでしょう。問題はRow.2に電流を流したときです。Cのキーが押されていますからCol.1に電流が流れる。これは当然です。さらに，Col.1に流れた電流はAのキーが押されている（AでRolラインとColラインがつながっている）ことによってRow.1にも流れ込み，さらにBのキーが押されていることによってCol.2にも流れてしまうのです。つまり2)のような経路が内部でできあがってしまうわけです。Row.2に電流を流したときにCol.2に電流が流れるため，Dのキーが押されたのだとコンピュータは判断してしまいます。押してもいないDのキーが入力されてしまう。これがISAOの原因です。

キーボードのチューンアップ

最近Oh!X誌上の広告に「このキーボードはひと味違う！　あなたのX68000のキーボードをチューンアップします」という広告が出ています。この広告を出している

図1　キーボードの模式図

図2　イサオ現象

㈱サイバーでは，前述のような事態が起きないよう，X68000のキーボード（PRO/PROIIを除く）に細工を施してくれます。

●ステージ1

キーボードのチューンアップは2つのステージに分かれています。ステージ1はキータッチに関するチューンです。X68000のキータッチはやや軽め（それでもX1turboよりは重い）で，キーの重さは押し始めからストロークの最後まで変わりません。これを少し重めでクリック感のよいものに交換するのがステージ1です。

新しいキーはストロークの最後で指にかかっていた力がスッと抜けるような感触をもっており，いかにもキーを押したという感じが指に返ってきます。これはLEDの付いたキーなど，構造上変更できないものを除いた94個のキーの下についているスイッチをすべて取り替えて実現されています。

使用感は明らかに従来のものより向上しています。キーのストロークは従来のものと同程度で浅すぎず深すぎず。そして最後のクリック感が小気味よく決まり，使っていると楽しくなってくるキーボードです。実は私個人はこれまでずっと，MZ-2500のキーボードが最高だと思っていたのですが，どうも自分の内部で美化してしまっていたようです。改めて比べてみると，ストロークと重さは同じ程度，最後のクリック感の分だけチューンされたキーボードのほうが気分がいいという結論に達しました。

これといった問題点は感じないのですが，キーはもう少し軽くてもいいのではないかと思います。これはX68000のキーボードに馴染んでしまったせいもあるのだと思います。指が弱体化しているのか，長い文章を書いていると少々辛いのです。以前はMZ-2500のキーボードで長い文章を書いていたのですから慣れの問題だと思うのですが。いずれにせよ，この使用感を文章でしかお届けできないのは残念です。量販店でデモでもしていただけると，一般ユーザーが直接触って確かめることができていいのではないかと思うのですが。

●ステージ2

続くステージ2は，ステージ1の変更を加えたあとさらに誤入力防止措置を施すというものです。図2-2のような経路がなぜできてしまうのかというと，本来RowラインからColラインへ流れるはずの電流が，Aキーの位置で逆流しているからです。つまり，ColラインからRowラインへの電流の逆流を防げば，誤入力を防ぐことができます。

電流を1方向にしか流さないものとしてダイオードがあります。そう，図3のような記号で表記されるアレです。三角形の向いている方向にしか電流を流さないダイオードを図4のように挿入すれば，仮にABCのキーが一度に押されてもAのキーのところで電流がせき止められ，誤入力が防止されるのです。

この処置はブラインドタイプでガシガシ入力する人には願ってもないチューンでしょう。かくいう私もそのひとりですが，幸いなことに私の指はキーを押し下げる場合だけでなくキーを離すほうにも素早さを発揮するらしく，2つはまだしも3つのキーを同時に押しているということはほとんどないようです。

それでもたまに，思わぬ場所で思わぬキーが入力されることがあるのは事実です。また，多くの人は自分でも気づかないうちに誤った入力をしているのではないでしょうか。ローマ字で入力を行っている人なら「あじ（AJI）」と入力したつもりが画面がスクロールしてしまったなんてこともあるでしょう。

なによりも，気持ちよくタイピングを行うにはキーボードの信頼性は重要です。誤入力はイライラのもとですから，ストレス解消のためのチューンアップという考え方もできます。ステージ2の29,800円の投資が，X68000に対する信頼性と愛着をぐんと大きくしてくれることでしょう。欲をいえば，CAPSキーなどの位置を変更してくれるようなカスタマイズステージがあればさらに嬉しいのですが。

最後にX68000のキーボードマトリックス図を，図5に挙げておきます。これを眺めながら3つのキーを押して，X68000を混乱させるというのも面白いかもしれません。

ステージ1：キースイッチの交換……19,800円
ステージ2：キースイッチの交換＋誤入力防止処理……29,800円
㈱サイバー　☎045(962)1447

図3　ダイオード

図4　ダイオードで逆流を防止する

図5　X68000のキーボードマトリックス

	Col. 7	Col. 6	Col. 5	Col. 4	Col. 3	Col. 2	Col. 1	Col. 0
Row. 0	CTRL	SHIFT	OPT. 1	OPT. 2				
Row. 1	全角	ヒラガナ	INS	CAPS	コード	ローマ	かな	
Row. 2	F6	F5	F4	F3	F2	F1	COPY	BREAK
Row. 3	7	6	5	4	3	2	1	ESC
Row. 4	U	Y	T	R	E	W	Q	TAB
Row. 5	K	J	H	G	F	D	S	A
Row. 6	,	M	N	B	V	C	X	Z
Row. 7	XF5	XF4	XF3	SPC	XF2	XF1		
Row. 8	0	HELP	登録	記号	F10	F9	F8	F7
Row. 9	DEL	HOME	BS	¥	^	—	8	9
Row. 10	R. DWN	R. UP	CR	[	@	P	O	I
Row. 11	UNDO	→	↑	←	]	:	;	L
Row. 12	(0)	(1)	(4)	(7)	(CLR)	_	／	.
Row. 13	(,)	(2)	(5)	(8)	(／)			↓
Row. 14	(.)	(3)	(6)	(9)	(*)			
Row. 15		ENTER	(=)	(+)	(−)			

（注）かっこの付いているキーはテンキーです。

PENGUIN INFORMATION CORNER
ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

タッチセンサー内蔵
VDB-1000
カシオ計算機

VDB-1000

カシオ計算機は，タッチキーを採用した対話型データバンクウォッチ「カシオ ビジュアルデータバンク VDB-1000」を2月25日に発売する。

この製品では従来の多機能腕時計の操作ボタンをなくしタッチセンサーを採用することで，指定の表示面に指で触れるだけで簡単に操作，入力ができるようになっている。表示面は64×32ドットで，アルファベット，カナ文字，数字，そして世界地図が表示可能。

情報管理機能のほうは名前と電話番号のほかに，住所の項目も入力できるテレメモ機能が用意されている。住所エリアは最大88文字まで入力でき，名前6文字，電話番号12桁で住所エリアを使用しない場合，最大120件まで記憶できる。

また，ワールドタイム機能では世界24カ所の時刻を各時刻帯の代表的な都市名で世界地図とともに表示。地図を左右にスクロールさせて見たい都市の時刻を探すタイムゾーン連続サーチ機能も搭載されている。

スケジュール機能はアルファベット，カナ文字，数字のメッセージ8文字と，月日時分で最大120件まで記憶でき，セットした日付，時刻になるとメッセージとともに電子ブザーが鳴って知らせてくれる。さらに，備忘録に使えるメモ機能，ミニ電卓感覚で使える8桁計算機能，ストップウォッチ機能などももちろん備えている。

価格は24,000円（税別）。

〈問い合わせ先〉

カシオ計算機㈱　☎03(3347)4811

普及価格帯19200bps
MultiModemV32S
アメリカン・テクノロジー・グループ

アメリカン・テクノロジー・グループは，MultiModemシリーズの新製品として「MultiModemV32S」を発売した。「MultiModemV32S」は通信規格CCITT V.32(9600bps)とMNPクラス5データ圧縮機能により，19200bpsのスループットを実現している。エラー訂正機能としてはCCITT V.42を搭載，また，コーリングシーケンスとしてCCITT V.25bisを装備，セキュリティ・コールバック機能なども搭載している。

価格は198,000円（税別）。

〈問い合わせ先〉

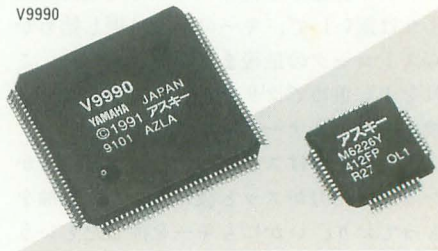
アメリカン・テクノロジー・グループ㈱

☎03(3503)4516

画面表示制御LSIを開発
V9990
アスキー

アスキーはヤマハと共同で，ワープロやパソコンなど家庭内の情報機器全般に利用できる，日本語表示機能を持ったVDP「V9990」を開発した。仕様と基本設計をアスキー，設計と製造をヤマハが担当しており，すでにヤマハからサンプル出荷が開始されている。サンプル価格は1個6,000円。

特徴は以下のとおり，

V9990

・最大32,768色同時表示

・高速日本語表示機能

漢字ROM「M6226Y」などを接続でき，CPUの負担を軽減し高速な日本語表示を実現

・テレビに最適な表示機能

オーバースキャン機能によりビデオ編集機などへの広範な応用が可能

・強力なグラフィック機能

高速描画(160ns高速ピクセル転送機能，漢字ROMパターン転送機能，直線描画機能），全方向スクロール機能（4画面ページング），スプライト機能（16×16ドット，125枚/画面，32枚/水平ライン），表示優先制御など

この「V9990」は両社がいままでに開発したVDPの技術，ノウハウをもとに開発されたもので，テレビ，パソコン用モニタ，液晶パネルなど主たる表示装置にこの1チップで対応することができる。

〈問い合わせ先〉

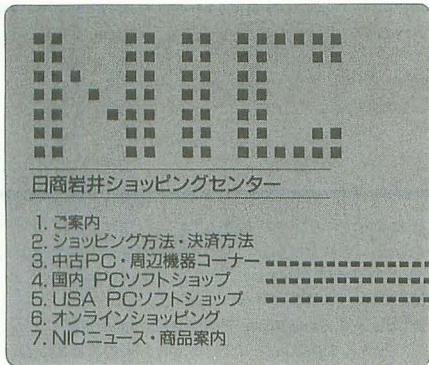
ヤマハ㈱　☎0539(62)4918

INFORMATION

中古パソコンの売買をパソコン通信で
日商岩井

日商岩井は，中古パソコンとその周辺機器の売買をパソコン通信で行う「中古パソコン通販ショップ」をNIFTY-Serveに開設した。

このサービスはパソコン通信を活用することで，より新鮮な情報の提供を実現し，中古売買で生ずるさまざまなトラブルをなるべく少なくしようというもの。

メニュー画面

購入希望者は，同社と富士通の共同出資会社であるエヌ・アイ・エフが展開しているネットを利用して，日商岩井ショッピングセンターの「中古パソコンショップ」上で希望する商品を発注する。同ショップは顧客の注文確認の電子メールを受信，指定銀行口座への振り込み確認後，商品を発送。約1週間で購入希望者の手元に商品が届けられるシステムとなっている（配送地域は全国）。

売る場合は，買いたい情報を見ながら電子メールボックスにその中古機の機種名や販売希望価格などの商品情報を入力すれば購入業者を斡旋してくれる。

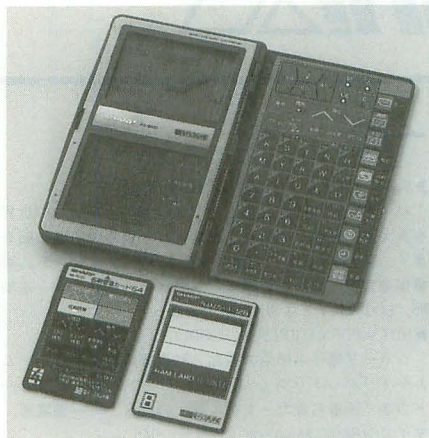
〈問い合わせ先〉
日商岩井㈱　☎03(3588)4598

電子システム手帳向け
名刺データインプットサービス
シャープ

シャープは凸版印刷とのタイアップによる「名刺データインプットサービス」を全国を対象に平成3年1月から4月末まで実施している。

このサービスはシャープの電子システム手帳用の名刺管理カード(PA-7C50/51)，またはRAMカード(PA-9C90/91)にユーザーが管理する名刺データを代行入力してくれるというもので，多忙なビジネスマンなどの利用を見込んでいる。

○入力料金：1口　100件　6,000円(税別)
○入力形式：名刺管理カード，ハイパー電子システム手帳「DB-Z」名刺管理機能に準じ，「個人名，個人名読み，電話番号，FAX番号，会社名，会社名読み，郵便番号，住所」の8項目
○申し込み先：〒553　大阪市福島区海老江3-23-50　凸版印刷㈱　トッパン名刺データインプットサービスセンター

〈問い合わせ先〉
凸版印刷㈱　☎06(454)3343
シャープ㈱　☎03(3260)1161, 06(621)1221

国内初のLAN総合展示会
LAN Expo '91
主催　マーコム・インターナショナル

国内初の本格的なLAN（ローカルエリアネットワーク）総合展示会“LAN Expo '91”が3月13, 14, 15日の3日間，東京ドーム・プリズムホールにて開催される。

同展示会は今回が第1回の開催であるが，LAN産業全体の活性化の動きを受け，日米の主要な関連企業91社が一堂に会することとなった。

○会期：3月13日から15日
午前10時から午後5時まで
（最終日は午後4時終了）
○会場：水道橋　プリズムホール（東京ドーム）
○出展製品：LAN関連ハードウェアおよびソフトウェア，コンピュータ関連商品（ワークステーション，パソコンなど），LAN関連の部品および部材

〈問い合わせ先〉
㈱マーコム・インターナショナル　LAN Expo事務局　☎03(3403)8515

C-TRACE CGコンペティション
キャスト

コンピュータグラフィクス(CG)に対する関心が日ごとに高まるなか，キャストではパーソナルコンピュータでもクオリティの高いCGができることをアピールするため，「C-TRACE CGコンペティション」を行う。

対象はキャストの製品を使用して作成されたCG静止画，アニメーションに限られ，以下の部門に分けられる。

・静止画キャラクタ部門
・静止画アート部門
・静止画産業デザイン部門
・アニメーション部門

募集期間は1月20日から3月20日までで，5月発売の本誌およびASCIIのキャスト広告誌面上で結果が発表される。

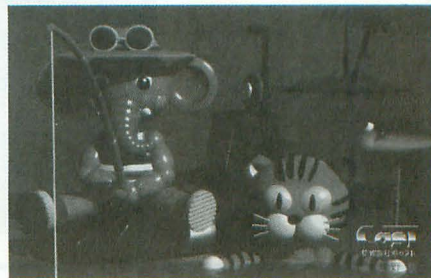
予定されている審査員は，
CGキッチンまざあぐうす代表
　長谷川　一光
C-TRACEユーザークラブ会長
東京芸術工学院講師
　塩沢　佐千子
超能力者
　玉手　峰人
イラストレイター
　井川　英雄　　（敬称略）
および，協賛会社代表
　㈱キャスト　スタッフ
となっている。

用意されている賞は，

グランプリ	1名
準グランプリ	1名
金賞	1名
銀賞	1名
銅賞	1名
ステゴちゃん賞	3名

協賛会社賞，各協賛会社特別賞
となっており，賞品は海外旅行，トランスピュータ，フルカラーフレームバッファ，キャスト商品券(15万円, 10万円, 5万円)より，グランプリから銅賞までの受賞者が順番に選択する。ステゴちゃん賞としてはキャストマスコットのぬいぐるみが，また，応募者全員に参加賞としてキャストオリジナルポストカードが贈られる。

詳しくはキャストの広告を参照のこと。

〈問い合わせ先〉
㈱キャスト　☎03(3705)1065

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。そろそろコタツとたわむれるのをやめて，来たるべき春に備えましょう。ちょっとの散歩もいい気分転換になりますよ。

一般

▶特集 '91新春パソコン業界座談会
ハードメーカー，ソフトハウスなど，パソコンの世界を支える「業界」の方々をゲストに，いろいろなお話をきく。X68000の生みの親であるシャープAVシステム事業推進室長・鳥居勉氏のインタビューも掲載。——編集部，LOGIN，1・2号，243-271pp.

▶HOT INFORMATION NEW PRODUCTS
シャープ電子手帳用の英和，和英辞典カード，ゲームカード「ハットリス」，名局観戦モード・棋力判定モードつき「囲碁名鑑カード第1巻」の紹介。——編集部，マイコンBASIC Magazine，2月号，94p.

▶MYCOM COME ON GOODS
プリンタ特集。ページプリンタから熱転写まで，印字サンプルを添えて紹介する。——編集部，マイコン，2月号，245-250pp.

▶エンジェルノートで104
NTT番号案内データベースへ直接アクセスできる専用端末機「エンジェルノート」の仕様紹介と試用記。——編集部，マイコン，2月号，316-317pp.

▶どこでもいくぞ日本パソコン百景
国内のハイテク映像作品を集めたビデオライブラリー，TEPIAライブラリーを取材。全自動ビデオ閲覧システムなどが採用されている。——フデヨシ＆カシワラ，ASCII，2月号，226-227pp.

▶パソコン'91年はこうなる
DOSやCPU，ノート型パソコン，ハードディスク，ソフトウェア，通信ネットなど,コンピュータをめぐる'91年の動向を大胆に予測する特集。——編集部，ASCII，2月号，234-256pp.

▶サイバーソン報告
バーチャルリアリティに関する報告討論会「サイバーソン」の模様のレポート後編。パソコンからバーチャル・スペースにアクセスできるサイバースペース構想の内容などを伝える。——野々村文宏，ASCII，2月号，321-328pp.

▶ソフトウェア特許
プログラミングの未来はこれでいいのかと称して，アメリカで問題になっているソフトウェアをめぐる特許の問題について，特許廃止の立場から述べる。——プログラミング自由化連盟，I/O，2月号，216-221pp.

MZシリーズ

MZ-1500（MZ-5Z001 BASIC）

▶χ・ρ（カイ・ロー）
自機，カイ・ローを操って画面内を動き回っている超圧縮エネルギー弾を打ち返して的にぶつける。ただのブロックくずしではないのだ。——SAMIQ＆OCTASS SOFT，マイコンBASIC Magazine，2月号，123-125pp.

MZ-2500（BASIC-M25）

▶Little Red Riding Hood－赤ずきんと10人の老婆－
いっぺんに病気になってしまった10人のおばあさん。赤ずきんちゃんはおばあさんの病気を直さなければなりません。しかし山の中にはオオカミがいて，赤ずきんをねらっています。——BMN，マイコンBASIC Magazine，2月号，126-128pp.

▶ザ・スーパー忍 〜OVER THE RAY〜
セガのミュージックのプログラム。——吉田知裕，マイコンBASIC Magazine，2月号，182-183pp.

X1/turdo/Z

X1シリーズ

▶BALL' S LAND
何者かによって動かされてしまったブルーボールを元の位置に戻してもらいたいわけである——栗林良樹，マイコンBASIC Magazine，2月号，155-156pp.

▶桂馬病
桂馬飛びしながらお金を拾い集める。——棚橋大介，マイコンBASIC Magazine，2月号，157-159pp.

X1＋FM音源ボード（要NEW FM音源ドライバ）

▶グラディウスII 〜浮遊大陸のテーマ〜
コナミのゲームミュージックプログラム。——桐畑厚宏，マイコンBASIC Magazine，2月号，186-187pp.

X1turboシリーズ

▶居た居た!! 何処に
2つのゲームフィールドの各キャラクタを動かして宝石をとる。——噂のショート・プログラマー，マイコンBASIC Magazine，2月号，160-161pp.

X1turboZ

▶LET' S PROGRAM
今月の宿題は「ブラックジャック」，turboZ-BASIC用のサンプルが掲載されている。SWAP文でシャッフルを行う，いかにもZ-BASICらしいもの。——藤本健，マイコン，2月号，260-267pp.

X68000

▶NEW SOFT
KLAX，イメージファイト，スライス，ラプラスの魔，D，Ryu，リングマスターIIの紹介。——編集部，LOGIN，1・2号，22-26pp.

▶最新ゲーム徹底解剖!!
続ダンジョン・マスター カオスの逆襲を攻略。移植中のプリンス・オブ・ペルシャを紹介。——編集部，L

参考文献
I/O 工学社
ASCII アスキー
コンプティーク 角川書店
テクノポリス 徳間書店
ポケコンジャーナル 工学社
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIC Magazine 電波新聞社
LOGIN アスキー

インゴ・ギュンターという人がいる。人工衛星などで調べた地球の環境を張り付けた地球儀を並べ「ワールド・プロセッサ」というインスタレーションを行った人だ。昨年春，日本でも開かれ，私も見にいった。暗い部屋に，中に照明を入れた地球儀が100個以上並んでいた。ただ作品を並べるだけではなく，展示場の空間自体を作品にしてしまうのがインスタレーションだ。各国のテレビの所有率，黒く塗った海，海流の流れ，衛星写真から起こした夜の地球，核兵器を持つ国，といった地球儀たち，数々の地球は圧倒的に見る者に迫り，しばし，メディア・アートという表現の可能性を信じる気になった。本書はインゴ・ギュンターの地球儀を56個集めた写真集である。しかし，写真にすぎない。自在に地球儀を回せる平面がほしいと思う。そいつは目の前にある。X68000だ。私が臨むのは，この56個の地球儀をX68000の中で回すことであり，メディアとしてのパソコンはこういう優れた作品をもって可能になるのだ。マルチメディアは時代の最大公約数である百科事典を入れるものではなく，こういった新しい文化を封じ込めるべきものなのだ。（K）

地球56の顔 インゴ・ギャンター 小学館刊
☎03(3230)5739 A4変形判 114ページ 2,200円

書籍の価格は消費税込みです

OGIN, 1・2号, 186-213pp.

▶Software Review
映画的手法を取り入れたゲームとして，ズームの「ラグーン」と「ファランクス」を取り上げて考えてみた。——編集部, LOGIN, 1・2号, 214-215pp.

▶X68000新聞
新着ゲーム大航海時代，ノスタルジア，KLAX，ブルトン・レイ シナリオ集2と開発中のファランクス，生中継68，パロディウスだ！，中華大仙を紹介。——編集部, LOGIN, 1・2号, 280-283pp.

▶とじこみスペシャル組み立て紙模型の友
世界のパソコンシリーズ第1回シャープX68000の巻(前編)。今回はX68000 SUPER-HDの1/5ペーパークラフトが付録。次回は専用ディスプレイの予定。——編集部, LOGIN, 1・2号, 付録・322-325pp.

▶GAMING WORLD
ワールドスタジアム，ダイナマイト・デューク，レインフォーサー，スライスを紹介。——編集部, テクノポリス, 2月号, 28-34pp.

▶攻略ファイト！
続ダンジョン・マスター カオスの逆襲を攻略。——編集部, テクノポリス, 2月号, 68-71pp.

▶How To Win
続ダンジョン・マスター カオスの逆襲，三国志II，銀河英雄伝説IIの攻略。——編集部, コンプティーク, 2月号, 142-153pp.

▶X68000SPIRITS
新着ゲームの紹介。ワールドスタジアム，エメラルド・ドラゴン，シュヴァルツシルトなど。——編集部, コンプティーク, 2月号, 252-253pp.

▶HARD PACK PAGE
SCSI内蔵の新機種「X68000 SUPER」を紹介。——編集部, コンプティーク, 2月号, 269p.

▶Hot Press
新着ゲームのルーシー・ショット，ザ・スーパーラスベガスII，エメラルド・ドラゴン，ノスタルジア，ソル・フィース，ラプラスの魔，栄冠は君に，中華大仙を紹介。——編集部, POPCOM, 2月号, 15-23pp.

▶ミュージック・パビリオン
THE真心ブラザース「どか〜ん」。X-BASIC用ミュージックプログラム。——編集部, POPCOM, 2月号, 163-164pp.

▶NAGDRV情報局
X68000用ミュージックドライバ「NAGDRV」関係の質問に，作者が答えている。——永田英哉, マイコンBASIC Magazine, 2月号, 88-99pp.

▶誌上公開質問状
X68000とアンプの上手なつなぎ方や，CZ-8PC4を使うためのプリンタ設定は？ ディスプレイ「CU-14GE」にビデオを接続するには？ など。——多田太郎, マイコンBASIC Magazine, 2月号, 91-92pp.

▶まる子対たまちゃん
昨日の友は今日の敵!? 落とし穴をほって相手を落とす。2人用対戦ゲーム。——溝口朋宏, マイコンBASIC Magazine, 2月号, 162-163pp.

▶ジャンケンおにごっこ
じゃんけんの法則にしたがって相手を捕まえる。おにごっこゲーム。——くえっ, マイコンBASIC Magazine, 2月号, 164-165pp.

▶ECSTASY 〜Red and Blue〜
5台の戦車とアイテムを使って相手の戦車を破壊する。2人用対戦ゲーム。——小野正明, マイコンBASIC Magazine, 2月号, 166-168pp.

▶ハードパンチャー
コナミのゲームミュージック。要NAGDRV。——M.H, マイコンBASIC Magazine, 2月号, 188-189pp.

▶MD版ソーサリアン「ペストの祭壇」
セガのゲームミュージックプログラム。要NAGDRV＋MT-32。——渡辺和広, マイコンBASIC Magazine, 2月号, 190-191pp.

▶GALAXIAN3 〜THEME OF GALAXIAN〜
ナムコのゲームミュージックプログラム。要NAGDRV(CM-64/MT-32＋CM-32P)——あんど, マイコンBASIC Magazine, 2月号, 192-194pp.

▶ゲームレビュー
続ダンジョン・マスター カオスの逆襲,ソル・フィースのゲームレビュー。——猪野清秀・MUNEPI♪, マイコン, 2月号, 190-197pp.

▶なんでもQ＆Aスペシャル
X68000とシャープ発売のアプリケーションをめぐる質問に答える。——シャープ株式会社液晶映像システム事業部第2商品企画部, マイコン, 2月号, 35-42pp.

▶小さくまとめて大きく使う「圧縮技術」の効能とは？
画像データ圧縮について。自前の圧縮プログラムを掲載。——MASUDA, マイコン, 2月号, 332-335pp.

▶AVプログラミング講座
バックグラウンドを含むさまざまな種類のオブジェクトを処理する方法について解説。——宮本親一郎, ASCII, 2月号, 329-336pp.

▶AV STRASSE
シャープから発売された光磁気ディスクドライブCZ-6MOIについてハードディスクとの速度比較を交えて紹介する。——仲田津宏, ASCII, 2月号, 361-364pp.

▶FREE SOFTWARE INDEX
主要な大手ネットワークに最近アップロードされたPDSのうち，主要なものをピックアップして紹介する。X68000用もエディタを始め，多数のプログラムが紹介されている。——編集部, ASCII, 2月号, 402-407pp.

▶長期ロードテスト
X68000 EXPERT IIの試用記。MIDIを使った音楽演奏に腰を抜かし，EDとワープロを使って四苦八苦しながら原稿を仕上げる。——編集部, ASCII, 2月号, 419-421pp.

▶DRVFMT＆DRVCPY
フォーマットとコピーのプログラム。動作中も他の処理ができるのが特徴。ただ多少の注意は必要。——(は), I/O, 2月号, 120-127pp.

▶EDITDISK
Human68k標準フォーマットのディスクの任意のトラックに対して，READ，EDITなどを行うディスクエディタ。——牛島之博, I/O, 2月号, 184-191pp.

▶SCSIボードCZ-6BSI
シャープのSCSIボードCZ-6BSIの紹介。SCSIの仕組みなどについても触れている。——市原昌文, I/O, 2月号, 202-205pp.

▶SOFT BOX
X68000用の日本語フロントエンドプロセッサ「FIXER Ver.4」を，製品添付の「ASK」と比較しながら紹介する。——花井志生, I/O, 2月号, 206-209pp.

▶GAME BOX
ソル・フィースとノスタルジアのゲームレビュー，パロディウスだ！と生中継68の画面写真を紹介。——市原昌文・編集部M・YRK, I/O, 2月号, 130-136pp.

ポケコン

PC-E500

▶ADDRIS
降りてくるブロックを積んでいき，となりあう列の高さを等しくして消すゲーム。——佐藤祐紀, マイコンBASIC Magazine, 2月号, 170-171pp.

▶BARATTA
魔術と剣を駆使して敵を倒す，超ハードなファンタジーアクションゲーム。——Touchable Mark V, マイコンBASIC Magazine, 2月号, 172-173pp.

PC-E500/E550/1480U/90U

▶EL-Sheet
表計算ソフト。99×18，24字までの大きさの表が扱え，ポケコンながらひと通りの機能を持っている。——英斗恋, ポケコンジャーナル, 2月号, 4-20pp.

▶水晶
水晶を取り，入り組んだ洞窟の出口まで辿りつけというパズルゲーム。——ビオラの安田, ポケコンジャーナル, 2月号, 83-85pp.

▶はみだしゲーム講座
16×16ドットキャラの表示をマシン語でするにはどうしたらよいかを考える。展開・表示ルーチンつき。——編集部, ポケコンジャーナル, 2月号, 74p.

レイアウトひらめき事典

DTPソフトを使って，紙面のデザインをする。案内状や社内報でもいい。ところが，いいデザインは難しい。本書は出版物のためのグラフィックデザインを1,000個以上並べた本である。見る人が見ればそこから自分のアイデアが練られるように巧妙に並べられている。本文レイアウトから見出しの処理，ビジュアルの表現まで。例は英語だが，素人の私でも触発される本だ。 (K)

レナード・コレン/R.ウィッポ・メッククラー著
渋川育由日本語版監修 河出書房新社刊 ☎03(3404)1201 A5判 137ページ 1,300円

JICCブックレット 情報社会の弱点がわかる本

人はコンピュータという希代の概念を得，あと先考えずにわわぁーっと突き進んだ。当然，法整備や倫理は遅れ，さまざまな問題が噴出している。わからないものはブラックボックスにして信じてしまおうという牧歌的信頼関係に依っているからだ。でもやばいものはやばい。何がどうやばいかについて，薄っぺらいブックレットに広く詰め込んだのが本書だ。詰め込んだためにわかりにくいところもあるが，読んで損はない。 (K)

名和小太郎著 JICC出版局刊 ☎03(3234)4621 A5判 79ページ 450円

書籍の価格は消費税込みです

QUESTION and ANSWER

X-BASICかC言語で，スプライトに円運動や楕円運動をさせる方法を教えてください。

岩手県　佐々木　信一

原点を中心として半径が1である円を単位円というのですが，円の中心座標さえわかっていれば，この円周上の座標は三角関数を使って求めることができます。中心座標(0，0)から角θの表す動径が単位円と交わる座標を（X，Y）とすると，

$X=\cos\theta$

$Y=\sin\theta$

と表せます。これは高校に入ってから基礎解析で勉強するものだと記憶しています。

ところでX-BASICでは角θをラジアンで記述することになっているので，角度に$\pi/180$をかけなくてはいけません。ですから中心座標を（256,256）として半径100の円を三角関数を使って描画しようとすれば，リスト1のようになります。

このプログラムでは角度を変化させるのにFOR～NEXTループを使っています（150行）。これを見ると角度iは0～45度の範囲でしか計算していないように見えます。しかし，ループの中でiを8倍しているので，実際にはiは0～360度まで8度ごとに変化していきます。これはX-BASICのFOR文にSTEPが使えないためです。

また，プログラム中でrxは横軸の半径，ryは縦軸の半径を表しています。rxに2/3（512/768を約分すると2/3）をかけているのは，画面の縦横の比率を調整するためです。rx＝ryで真円，rx>ryで横に平べったい楕円，rx<ryで縦に平べったい楕円となります。

サンプルでは四角いパターンがぐるぐると回っているだけですが，これをちょっと工夫して，パターンを円の接線に対して垂直に表示させる方法を考えてみましょう。

X68000のスプライトはハードウェアでの回転機能を持っていません。ですから，あらかじめ回転させたパターンをスプライトに定義しておいて，必要なときはそのスプライトを表示するようにすればいいでしょう。円の1/4の部分について回転パターンを用意しておけば，その他の部分についてはパターンの反転で対応できることもあります（親切にいうと，スプライトパターンが上下左右に対称な場合です）。

しかし，たとえば，左半分が赤，右半分が青のスプライトパターンがあったとします。このパターンを30度回転させたパターンをY軸に対して反転させるとどうなるでしょうか。赤と青が逆になってしまいますよね。XとYをともに反転した場合はちゃんとなることもおわかりでしょう。このように単純な反転で対応できない場合でも，すべてのスプライトパターンの半分を用意すれば足ることになります。

ぼくのX68000ACE-HDのハードディスクは友人のX68000ACE-HDよりも読み込みが遅いのです(すごく)。友人に聞くと，ディレクトリやファイルの並べ方などが悪いといわれました。しかし，方法がわからな～い？ソートというコマンドで並べ替えてやった！　と思ったら，また元どおりになっているではあ～りませんか。どうか並べ替える方法を教えてください。

大阪府　巻山　尚

確かにディスク上のファイルの位置によってアクセス速度が違うこともあります。たとえばフロッピーディスクに10Kバイトのファイルを100個作ったとして，ディレクトリの先頭のファイルと100個目のファイルとで，どれくらい読み込み時間に差があるというのでしょうか。コンマ何秒かでしょう。よく使うプログラムを4重5重に階層構造になっているディレクトリに置けば，読み込みが遅くなるに決まっていますが，基本的にファイルやディレクトリの並びが読み込み時間に大きな影響を与えるとは思えません。すごく遅いとまで感じるのですから，なにか別のところに原因があるものと考えていいでしょう。

断わっておきますが，SORTコマンドでディレクトリやファイルの並びを変更することはできません。自分でプログラムを作れば一番勉強になるのですが，市販ツール（FileProfessorなど）やPDSなどに，ディレクトリやファイルの並びを変更できるものがあります。しかしそれでは解決にはなりません。

巻山さんはX68000を使いだしてから1年だそうですから，その間にハードディスクにアクセスした回数は相当なものでしょう。アクセスした回数と読み込み時間の遅さは直接関係ないのですが，使った期間が長いだけ，これから述べるような操作をしたことも多かったと思います。

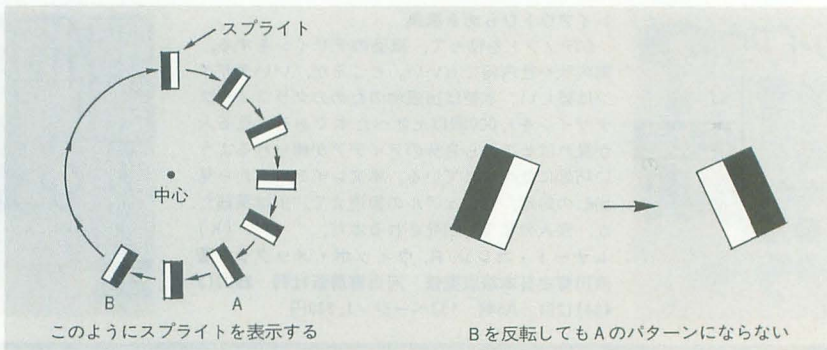
たとえばあるディレクトリにAという5Kバイトのファイル，Bという10Kバイトのファイルを置いていたとして，いまAがいらなくなったので削除したとします。その後フロッピーなどから，このディレクトリにCという15Kバイトのファイルをコピーしたとします。このときDOS（ディスクオペレーティングシステム）は15Kバイトが格納できる空きエリアを探します。しかし，連続した15Kバイトの空きエリアがなかった場合は，先ほど削除したAの空き領域に5Kバイト，Bの後ろに残りの10Kバ

●リスト1

```
10 /*
20 /* サンプルプログラム
30 /*
40 int x,y,rx=100,ry=100
50 dim char spl(255)
60 for i=0 to 255
70   spl(i)=i mod 16 /* スプライトデータ設定
80 next
90 screen 1,3,1,1     /* 画面モード 512*512
100 sp_init()
110 sp_clr(0,255)
120 sp_disp(1)
130 sp_def(0,spl,1)   /* スプライト定義
140 /* 円運動
150 for i=0 to 45
160   x=cos(i*8*3.14#/180)*rx*2/3+256
170   y=sin(i*8*3.14#/180)*ry+256
180   sp_move(0,x,y,0)
190 next
200 /* ループ
210 goto 150
```

図1

このようにスプライトを表示する

Bを反転してもAのパターンにならない

イトといったふうにデータを格納していきます。こういった作業を重ねるうちに，いつかハードディスクの中には細切れになったファイル（この例ではC）がたくさん存在するようになります。これはかなり悲惨な状態です。

Cを読むときには，まず最初に5Kバイトを読み込んで，Bの10Kバイトを読み飛ばして残りの10Kバイトを読み込みます。これは週刊誌などで漫画を読んでいると途中で広告が入ってたりしたとき，その部分を読み飛ばす動作とよく似ています。つまり，読み飛ばす時間だけスムーズにデータが読み込めないのです。ひとつのファイルを読み込むのにデータがあちこちに点在しているので，シーク時間（ヘッドの移動時間）が長くなり，結果として読み込みに時間がかかるのです。巻山さんのハードディスクには細切れになったファイルがたくさん存在しているのではないでしょうか。

細切れになったファイルを元に戻すには，ハードディスクの中身を整理し一度フロッピーにバックアップしたあとに，ハードディスクのファイルをすべて削除して，それからフロッピーにバックアップしたファイルをハードディスクに転送するよりしかたありません。ハードディスクのバックアップにはCOPYALL（Human Ver.2.0以降）を使うと快適です。バックアップする際には，手元にブランクディスクをたくさん用意しておきましょう。

ディスクに十分な容量があればDOSは高速にアクセスできるようにディスクの磁性面を使います。しかし，残り容量が少なくなってくると一気にツケが回ってきて，フロッピーよりも遅くなることもあります。これはディスクが大容量化すればするほど顕著です。ディスクを高速に保つにはメンテナンスを怠らず，常に十分な空きエリアを確保しておくことが重要です。

X-BASICでRPGを作っているのですが，マップ用の変数は1次元配列でできるのでしょうか。僕は2次元配列を使っていますが，ファイルを読み書きするのが遅すぎてどうしようもないんです。

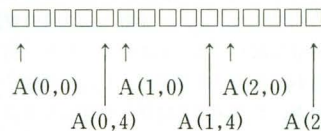
こんな方法でやっています。これは遅くてしようがありません。

```
10 for y=0 to 31
20 for x=0 to 31
30 map(x,y)=fgetc(file)
40 next
50 next
```

1次元配列を2次元配列として使う方法をぜひ詳しく教えてください。よろしくお願いします。　神奈川県　野沢　功一

1次元配列を2次元配列と見立てて使うことはできます。たとえばA(2,4)という2次元配列があったとすると，この配列は下のように図示できます。

A(0,0)→□□□□□←A(0,4)
A(1,0)→□□□□□←A(1,4)
A(2,0)→□□□□□←A(2,4)

これをA(0,4)，A(1,4)で改行せずに，横につなげて書き換えてみると，

のように表せます。こうしてみると，2次元配列が1次元配列で表せることがよくわかると思います。

一般に2次元配列A(M，N)の要素数は，(M＋1)×(N＋1)ですから（注：CZ-8FB01などで，OPTION BASE 1が指定してあるときはM×N），上の例のA(2,4)の要素数は15個です。ですから1次元配列B(15)でA(2,4)を表すことができます。そしてA(X，Y)と表す代わりに，B(X×(N＋1)＋Y)とします。すなわち，A(0,0)がB(0)，A(1,0)がB(5)，A(2,3)がB(13)と対応することになります。

同様に3次元配列A(M,N,O)も，1次元配列B((M+1)×(N+1)×(O+1))で表すことが可能で，A(X,Y,Z)はB(X×(N+1)×(O+1)＋Y×(O+1)＋Z)とすることができます。

このように1次元配列で表せることがわかったら，質問にあったプログラムは2次元配列mapが32×32の要素を持つ2次元配列なので，下のようにすっきりと書き直すことができます。

```
10 dim int map(1024)   /*=32×32
20 ai=fopen(file)
30     fread(map,1024,ai)
40 fclose(ai)
```

いままで2次元配列を使ってmap(x,y)としていたところは，map(x*32+y)とします。

しかし2次元配列を使っていても，プログラムの変更によって実行速度の改善を図ることができます。野沢さんのプログラムが遅いのはファイルからデータを1文字ずつ読み込んでいるからです。これはX-BASICが遅いこともありますが，実はコンパイルしてもそれほど速くなりません。たとえばS-OSなどは，ファイルの読み込みはレコード単位（256バイト単位）で行われています。読み込まれたデータは一度バッファに読み込まれ，そこから必要なバイト数だけ転送するようにしているのです。

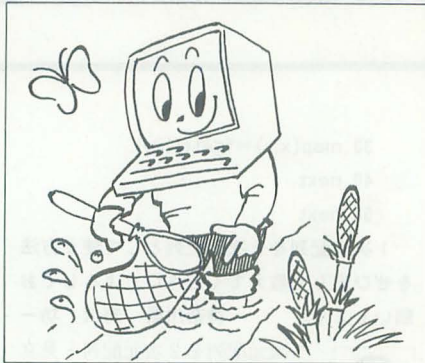
ですから野沢さんのプログラムも，1回の読み込みで1024バイトを1次元配列にとりあえず読み込んで，それから2次元配列に格納していくようにすれば実行速度が格段に向上します。

```
10 dim int map(31,31),buffer(1024)
20 ai=fopen(file)
30     fread(buffer,1024,ai)
40 fclose(ai)
50 i=0
60 /*  2次元配列に格納する
70 for x=0 to 31
80     for y=0 to 31
90     map(x,y)=buffer(i)
100    i=i+1
110    next
120 next
```

のようにしてみてください。（影山裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

今月のハガキは時期外れの年末ネタとか，お正月ネタとか，クリスマスネタとかが多いですが気にしないでください。なにぶん1月号のハガキなもんで。中東は大変なようですが，やっぱりそれに対するハガキもきてたりします。

◆今後のおまけディスクのタイトル名を提案する。豆まきPRO-68K，新春PRO-68K，創刊9周年記念PRO-68K，不意打ちPRO-68K，夏休みPRO-68K，暑中見舞いPRO-68K，かえで（もみじ）PRO-68K，サンタの贈り物PRO-68K。ふう，こんなもんでしょ。おまけディスクは“起”，“承”ときたから次の“転”はとんでもないものにしてほしいです。それと，ゲームミュージック（CD）のコーナーもほしい。たのむ。

白井　達広(17)愛知県

不意打ちPRO-68Kというのはなかなか。

◆1月「謹賀新年PRO-68K」，2月「建国記念PRO-68K」，3月「合格記念PRO-68K」，4月「入学記念PRO-68K」，5月「ゴールデンウィークPRO-68K」，6月「創刊記念PRO-68K」，7月「夏休みPRO-68K上」，8月「夏休みPRO-68K下」，9月「敬老記念PRO-68K」，10月「スポーツPRO-68K」，11月「勤労感謝PRO-68K」，12月「クリスマスPRO-68K」，という具合に毎月ディスクを付けてください。

後藤　秀樹(15)福島県

これは結構オーソドックスな感じ。

◆あけましておめでとうございます。1月号のディスクはとてもよかったですね。ところで，今度のディスクは4月号の予定だそうですね。ということは，ズバリ4月号のディスクはいろいろなツールが入っているとおもいきや，実はなんにも入っていなかったという「エイプリルフールPRO-68K」ではないでしょうか。そして，この次に出るディスクは，8月「残暑お見舞いPRO-68K」，10月「芸術の秋PRO-68K」，12月「クリスマスプレゼントPRO-68K」でしょう。

小川　保彦(18)東京都

ディスクは5月号になるんです。ごめんなさい。

◆12月のある日，Oh!X編集部からグレーの封筒が届きました。しばらく考えても思い当たることがないので，こわごわ開けてみると，なんと電脳倶楽部特製豆しぼりが当選し送られてきたのでした。本当に感謝しているのですが，同封の手紙に記されていた「お送りしました品物につきましてご使用になった際のご意見，ご感想などがありましたら当編集部までお寄せください」の文章が気になりました。私に豆をしぼれというのですね？

佐野　法之(19)東京都

「豆しぼり［名］小さい丸い形を一面にあらわした，しぼり染め」なんですけどね。

◆あ〜あ，クリスマスはひとりぼっちでつまらないなあ。こんなときはボクのロクハチに赤い服を着させて，白い髪とひげをつけて，♪ロクハチがサンタクロース，値が高いサンタクロース♪，などといっているうちに正月が来てしまう。でも，アルバイトもやってお金を稼がないといけない。やりたいこといっぱいある。あ〜あ（ためいきばっか），たいへんだァー！

志賀　宗一(17)愛知県

クリスマスはお仕事してました。

◆私は30歳である。サラリーマンである。1児のパパである。一応，部下と呼べる者もいる。1カ月の小遣い……，15,000円である。やっとX68000 PRO-HDを手に入れた。X68000 PROIIが出たおかげである。1カ月の小遣い……，15,000円である。毎日残業である。小さい会社だからフレックスタイムなんてない。X68000 PRO-HDのスイッチは午前0時の前に入れたことがない。でも，うれしい。XCのマニュアルと首っ引きである。プリンタがほしい。1カ月の小遣い……，15,000円である。プリンタが来た。プリンタのスイッチは午前7時の前に入れたことがない。X68000 PRO-HD＋プリンタ＝￥？？？である。1カ月の小遣い……，5,000円である。

福井　司(30)北海道

ローンのせいで1カ月の小遣いが5,000円に？「でも，うれしい」というのに実感がこもっていますね。

◆先日，パソコン通信“NIFTY-Serve”に入会した。どんなことをやっているのかと，ちょっとFSHARPというフォーラムをのぞいてみた。次にNIFTY-Serve入門という本に書いてあるメッセージを早く読む方法を試してみる。メッセージが516件あったが気にせず一挙にダウンロード。20分経過して途中で断念。次はリアルタイム会議室とやらをのぞいてみる。すると，「516件を一挙にダウンロードしている無謀な人がいる」と話していた。私はダウンロードしたばかりのデータもなくしちゃう無謀な初心者なのであった。

森本　俊昭(27)千葉県

まあ，なんでも初心者っていうのは無謀なものですから，いいんじゃないでしょうか。

◆12月26日，浜松アリーナ。新日本プロレスにおいて，かのルー・テーズの勇姿を見られたことがとても感激です。74歳になってもいまだに栄光のバックドロップは健在でした。もう1回，VTRでどうぞ。えっ！　Oh!Xにはまったく関係ないって？

段　宏太郎(19)兵庫県

ルー・テーズってまだプロレスやっていたんですか。そういうのは馬場だけかと思っていました。

◆シャープがぼくのための活動をしてくれるのかと思ったら「おみこし」だった。

浪越　孝宏(18)和歌山県

……名前を見て納得。

◆胃が痛い……。ふと思ったのだが，この胃痛は4次元人が私たちを苦しめるために，われわれの胃をこねくりまわしているのでは……。とすると，発熱は火であぶっているのかなあ……。桑原々々……。

伴　哲也(18)京都府

そんなこと考えているとますます胃が……。

◀正田　誠宏　兵庫県
イラストに書かれているモンスターはコミカルな感じ。名前が田子作とかいいそうですけど。鍬で攻撃したりして。

◀平　智征　神奈川県
不思議の国のアリスって実はグロテスクな雰囲気もありますよね。メルヘンメイズはメルヘンチックなだけですけど。

◆この間，防衛大学の2次（面接＋身体検査）に行ったら，身体検査のすごいことこのうえなかった。あれはキューキョクだった。なにせ4回も着たり脱いだりで，さらに血まで抜かれた。ヴー。1991年，いったいぼくは何をしているか考えると結構落ち着きません（浪人はいやだー）。が，1991年もがんばっていきましょうね！　仁泉　大輔(19)福岡県

文面からすると実は普通の身体検査のような気もしますが，ほかにもいろいろあったんでしょうね。

◆1月号の69ページのイラストはSX姫とネズミ男と14人の小人では？　宮越　良幸(18)神奈川県

正解！

◆久し振りだ。Oh!MZ……，おや，Oh!Xか。改めて，Oh!Xを読むのは久し振りだ。X68000 ACEの私ももはや1Mバイト RAMボードがほしいと思いだした。X68000はなぜか愛着がわくようだな。X68000 SUPERもいいけど，X68000 ACEもなかなか味があるぞ。みんなX68000 ACEとX68000 SUPERを交換しないか。しないか。そうか。　新屋　武弘(16)兵庫県

あんまりしないでしょうね。やっぱり。

◆表紙の順番を決めるのに，1年分引っ張りだして，ああでもないこうでもないと23分も考えてしまいました。やはり，'90年いちばんの出来事である（これが載るころには過去形になっていてほしい）イラクのクウェート侵攻に対してサウジアラビアに派遣されたアメリカ兵を思わせる10月号が印象に残りました。　木下　卓也(19)埼玉県

見直してみると確かにそういう感じ。

◆表紙BEST12，

1位8月号　どう考えたって1位でしょう。いちばんきれいで映画のラストシーンをほうふつとさせるような……。

2位5月号　影の部分を正面にしたところがいい。夕日が趣深い色を出しているのもいい。

3位9月号　11月号もそうだったけれど，作者の方はブタさんが好きなんですね。最も未来的でいいです。

4位10月号　疲れはてた旅人ってやつでしょうか。さりげなくバックの建物が崩壊しているのがいいです。

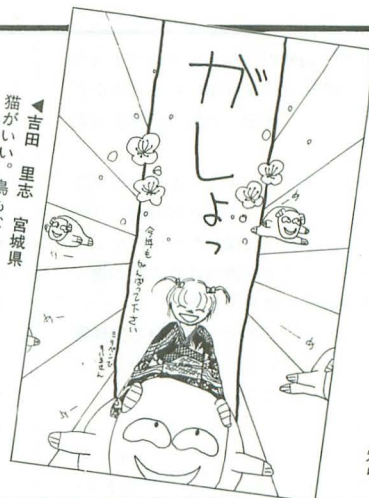
5位7月号　表情なく働くロボットと索莫とした風情が，聖飢魔IIのアノ歌を思い出させるのですが。

6位1月号　おっと6位に手描きが入っている。この世の終わりみたいでいいです。

7位11月号　もっと上位に入れたかったのですが，ところどころに入った文字が雰囲気を崩してしまっているような。

8位12月号　インドっぽい塔とかゾウ(?)はいいのですが，似たような色が多くてインパクトがないような……。

9位6月号　塔がとってもいいのに，UFOが多すぎて台無しにしていると思います。数を減らせば上位。

◀吉田　里志　宮城県
猫がいい。鳥もなんとなくいい。もちろん，女の子もかわいいです。でも，鳥の単純なタッチがいちばん好き。

◀小林　洋美　東京都
見ればわかるとおり，年賀状です。今年はなんと未年なんですよ。皆さん知ってましたか？　知らなかったでしょう。

10位4月号　背景の色がちょっと。もっと暗くすれば，絵が浮き上がると思います。あまり好きじゃない。

11位2月号　一瞬，スペースハリアーか？　と思ってしまうような絵。でも，CGにはかなわなかった。

12位3月号　全部の中でいちばんのっぺりしていると思う。こんなことシロウトが言うなって感じですけど。

以上，一般人のどうしようもない見解でした。　小笠原　洋(16)東京都

1つひとつに詳しい感想をありがとうございます。これでそれぞれの絵も浮かばれることでしょう。

◆1990年の本誌の表紙をよかった順に書いてください。えー，といいつつ狭い部屋にOh!Xを並べて「やっぱり，いちばんは1月号かな。いや，8月号も捨てがたいかな」などと真剣に悩んでいたのはぼくだけだろうか（このネタのハガキはたくさん来るだろうな）。話は変わりますがLIVE in '90でHELLOWEENの曲が載ってから，HELLOWEENのCDをレンタルで借りてきたのですが，なにか気に入ってしまいました（変な文章）。プログラムはまだ打ち込んでいないけど。　阪本　泰博(20)大阪府

Oh!Xを並べて見ているうちに眠くなって，ベッドにしてしまわなかったでしょうか。

◆クリスマスである。正月である。帰省の季節である。これから，1カ月近くX68000とお別れだ。今年のクリスマスは気のない女の子に誘われた友人の逃げに使われてすっかり悪者だ。「クリスマス，一緒に過ごしてくれませんか」，「友達と約束あるから」，だってさ。どちくしょお。こちとら工業大学で女の子のおの字もないんだぞ。　杉崎　充典(20)北海道

その友人に紹介してもらうとか。

◆あけましておめでとうございます。X68000と出会ってOh!Xなどとにらめっこを続けていますが，なかなか思いどおりにならないというのが現実。毎年，パソコンで年賀状を作ろうと決心しながら，結局プリントゴッコのお世話になってしまいました。来年こそは……，と早速目標に掲げました。X68000が私の手足や頭になるのはいつのことやら。ありがたいのは私の性格が決して短気じゃなかったことです。本年も宜しくお願いします。　矢吹　準子(30)福島県

今年はプリントゴッコを買おうと思ったのですが，結局手で書いてしまった。

◆昨年の11月に年末ジャンボ宝くじの2等1千万円が当たった夢を見ました。しかも，番号まで覚えているというリアルさ！　友人が「これは神のお告げだ」といい，その番号を探そうとしましたが，面倒なのでテキトーに買った。すると，やっぱり当たらなかった。しかし，1等の当たり番号に見覚えが……。今度は買う場所の夢が見たい。　加藤　信之(20)東京都

もし，また夢を見たら連絡ください。おすそわけを……。

◆1カ月ほど前からCRTの調子がおかしくなりました。最初は「パチッ」という音とともに画面に白い点が光るというものでした。そして，常駐プログラム「あられ」とでも名づけようかと思いながら，そのまま使っていました（時々本降りがきてなかなかリアル）。そしたら，ある日突然「バチッ」という音とともに画面に稲妻が走りました。それでも，「かみなり」も常駐したかな（どんなデモのものよりリアルだった。スピーカーもないのに音までつけて），などと使っていました。最近ではパラパラまんがのように表示してくれます。CU-14FIって芸達者だね。　狩野　太郎(17)富山県

そんなこといってないで修理に出さないと危ないですよ。

◆電脳倶楽部特製鉛筆ありがとうございました。「ものたりないH」のHB，「ちょっとH」のH，「けっこうH」の3H，「かなりH」の5H，「とってもH」の7H，「すごくH」の8H。8Hなんかわらばん紙が破れます。この鉛筆たちを持って受験に行くつもりです。　越智　亮(18)島根県

いやがらせに8Hで書くとか。そんなことしても落ちるだけか。

◆今日，エンベロープに収めていない5インチ2HDのディスクをストーブの前にうっかり置いてしまった。気づいたときにはディスクが少しゆがんでいたけど，磁性面は大丈夫だったようだから，ロードしたらちゃんと読めて，ちゃん

とファイルが取り出せた。なんとそのディスクは今月のOh!Xのディスクを解凍したディスクだった。　大田　哲矢(18)神奈川県

解凍ねえ。

◆カレンダー.Xの年号を1年にしようと思い立った。マウスをクリックすること1990回，うーん感動。1年12月25日は火曜日だったのか。ついでに，紀元前はマイナス表示されるかと思ったのに表示されん。一度お試しあれ。　木全　和茂(17)愛知県

やろうと思う人はいても，実際にやった人はあまりいないかな？

◆初めてアダルトゲームなるものを買った。elfのレイ・ガン。悪くはなかったが，つめが甘いなという感じがした(絵が16色)。ブルトン・レイを買うつもりで行ったのになく，なぜかレイ・ガンになってしまった。こまったもんだ。　五十嵐　博幸(20)岩手県

名前が似ているから間違えて買った人も……，いないですよね。

◆年末だけという約束で八百屋を手伝っています。みかんの箱で壁を作って，その後ろにはX68000が置いてあります。時給が安いんだから，X68000を置かせろという交渉の賜です。　下田　達也(23)三重県

わがままなアルバイター。

◆ぼくの学校ではPC-9801ユーザーがやたらに多い。しかも，ド高いPC-9801RAを持っているやつから，PC CLUBを持っているやつまで……。PC-8801も多く，昨年のクラスでは男子の中で(26人中) 8人くらいがPC-8801SR以降の機種を持っていた。はっきりいってNEC天国である。X68000ユーザーは2年生の中ではぼくひとりみたいである。悲しい。PC-9801ユーザーはことあるごとにソフトの数を自慢するやつが多い(そうでない人もいる)。しかし，「PC-9801はビジネスにだって使えるし，ゲームも多い」といっているやつらは，PC-9801をファミコン並みにしか使っていない。ぼくは多額のX68000への投資を無駄にしないように，X68000を使いこなしていきたいと思う。しかし，C compiler PRO-68Kは高校生にとっては高い。買えない。悲しい。　吉田　順一(16)愛知県

まあ，ぼちぼちお金を貯めて買うしかありませんね。やっぱり。

◆こんにちは！　年の瀬となりました。今年を振り返り，X1の稼働時間と私のマシン語習熟度のともなわぬことにアセっております。さて，今年1年というと猪木信者の私は感動の嵐で，これが猪木イズムだと，あらためて確信しているしだいです。「1･2･3　ダーッ」の掛け声も，あの9月30日横浜アリーナで実現。よっしゃ，よっしゃの1年でした。　佐藤　久彰(21)茨城県

猪木さんはますます元気なようですね。

◆バイトをしています（冬休み中ずっと)。内容はひみつ。ほしいブラウスもちょっと我慢しています。理由はひみつ。甘いものも控えています（これはダイエット……)。弟が今年の異常気象は私のせいだといいます。親知らずがにょきにょき出てきています。痛いよおう。ぐしぐし。　岩瀬　貴代美(19)福岡県

よくわからない。

◆空手初段を取りました。もう，こんなハガキの1枚や2枚は簡単に引き裂くことができます。強いでしょ。　廣田　政則(16)北海道

強い！

◆KLONDIKEに燃えてしまった。それで，普通のトランプでやってみたら，つまらなかった。　種村　信一(17)三重県

面倒臭いからかな。

◆友人Uは遮断機を折った。遮断機が降りてきているのに無理に突っ込んだら，完全に遮断機が降りてしまい，しかたがないので折っていったという。ちなみにLagoonを「ラグーン」と読んだのも彼である。　円福　貴光(17)愛知県

嵐のような友人を持って幸せですね。

◆あけましておめでとうございます。早速ですが今回の付録はすごい。毎日楽しんでいます。しかも，ビジュアルシェルに組み込んでくださっているから，非常に使いやすく，われわれ初心者にはありがたく思っています。4月号が楽しみ。　藤倉　朝賀(64)徳島県

どうもどうも。

◆この間，学校で献血をしたのです。たった200ミリリットルですが，これでも一応勇気がいると思います。学校全体で200名弱の人たちが小さな勇気を出して献血をしました。ぼくもそのうちのひとりですが，やったあとの品物がなかなかよかったなあと思ったから，これからもやろうかなあと思う今日このごろである。　鈴木　宏幸(17)愛知県

何をもらったんですか？　コーヒー牛乳？

◆VS2の画面は，いままでほとんどVS画面を見たことがなかったぼくにとって，とても新鮮でした。でも，ディスク4枚というのは痛かった……。　藤田　宏志(16)滋賀県

また，生ディスク用意しといてくださいね。

◆やっとの思いでCM-32Lを手に入れました。その音質のいいこと，「Mu-1」のデモ曲や，「ジェミニウイング」など，もう涙ものです。ここまでの道程は長かった。半年前，車で「Mu-1」を買って帰る途中，ガードレールに車をぶつけ，フロントバンパーが大破。そのため音源を買うための6万円はすべてパー。そのおかげで半年間もMIDIボードがささったままという，間抜けな状態になってしまったのです。しかし，それもこれで終わり。これからは十分に楽しめます。　三浦　裕行(26)神奈川県

本当に“やっとの思い”ですね。

◆先日，コミケットというところへ初めて行きました。電車賃込みで2万円もあればすむだろうと思っていました。帰りに秋葉原にでもよってソフトを買うつもりだったので，コミケに注ぎ込めるお金は1万円前後のはずでした。ところが，どっこい1万円どころか，2万円もぱぱぱっと使ってしまい，そのうえ友人に借りた1万円も注ぎ込んでしまったのです。帰りの電車賃すら使いはたした私は友人に頼み込んで，どうにか家に帰りました。しかし，この友人というのがくせもので，コミケに場なれしてないやつを誘っては衝動買いさせ，金の足りないところにつけ込んで高利で金を貸すのです。ひどいやつですよ。まったく。P.S.私はあきれた。注ぎ込んだ3万円，すべてH本だったなんて……。　足立　義宗(16)埼玉県

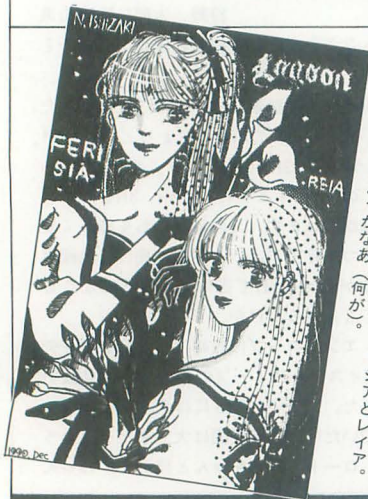
うーん，どっちもどっちという感じも。

◆うちのいなかはオグリキャップに栄誉賞を贈ってしまうバカな県です。ちなみに岐阜県です。成人式で帰省した1月15日にはオグリの引退セレモニーが笠松競馬場であったんですが，人口2万人そこそこの笠松に3万人もの人がやってきて車があふれかえり，名鉄が笑いました。馬主の小栗さんも笑いが止まらないことでしょう。　箕浦　真(20)東京都

夢があっていい県じゃないですか。

◆昨年末にアスキーネットに入会しました。しかし，我が家の電話はX68000から遠いところにあるので，仕方なくPC-9801NSを電話のそばまで持っていって通信しています。これからは寒くなることだし，暖かい自分の部屋に電話を引いて通信したいと思う今日このごろです。　椎橋　茂(25)東京都

長ーいケーブル買えばいいんじゃないでしょうか。隣家の電話に届くくらいのやつ。

◆29,800円でさらしものになっていたX1turbo

◀石崎　伸枝　栃木県
初めはラベック姉妹かと思ってしまいましたが，ラグーンのイラストですね。フェリシアとレイア。どっちにしようかなあ（何が）。

◀加藤　富盛　愛知県
いやあ，いいですねこの雰囲気。ブルトン・レイとはちょっと異なる感じかもしれませんが，絵のタッチが「適当な」ようでいいです。

Ｚ（中古）を買ってきた。キーボードをネジも外さずに開けようとしたらしく，プラスティック製カバーが半分浮いていた。マウスがなかったのでXⅠturboに使っていたやつをつないでいる。前の持ち主の顔が見てみたい（大阪のソフマップに売ったヤツ出てこい）。まあ，そのおかげでXⅠturboZユーザーになれたけど。これからかわいがってあげるからね。それにしてもアナログ4096色はすごい。これでセーブがもっと早かったら……。ほかにもOh!MZ1987年２月号の隠れ機能とか……。こうなったら，Z-BASICを買うしかない(いまさらどこに売っているねん)。しかし，XⅠturboはディスプレイ(CZ-602D)を取られてビデオ出力で使うはめになった（手でケーブル差し換えて使っているが面倒臭くて。でも，切り替え器を買う気はないし……）。

岡田　篤志(20)兵庫県

浦島太郎のような心境ですかね。

◆うちのおばあちん（ちゃん）は枝毛のことを「またげ」という。おじいちんは鉛筆のことを「いんびつ」といい，近くのバーではレーザーディスクのことをレザーデスクと書いてあった。ちなみに，うちの大学の光学の先生はルビーレーザーのことをルービーレーザーと書いている。

稲松　清澄(22)神奈川県

「いんびつ」というのは，なまっているだけでは？

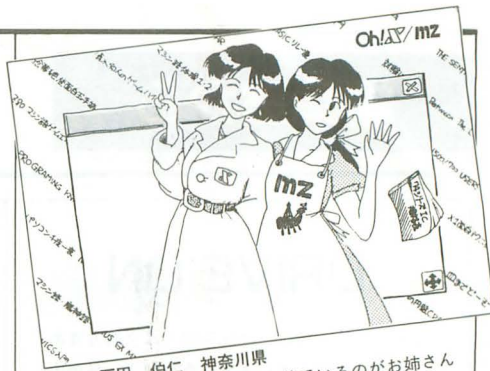
◆最近，ぼくのX68000がにわかに活気づいてきました。ゲームのほかにもMPUの使用法があることをほうふつとさせてくれます。いろいろやっています。初めてコンパイルをRAMディスクでしました。あっという間でした。ゲームを作っています。マップの上を勇者が高速で走っています。20Mバイトのハードディスクがはちきれるううう。

天満　一裕(19)大阪府

いいのができたら送ってください。

◆１年前に友人にXⅠturboを売り飛ばして以来，久々にOh!X買いました……。なぜか？　明後日にX68000PROⅡを買いにいくよーん，です。実は小生，写真をやっていまして将来はプロに！

とひそかにたくらんでいるのですが，いままでに撮ってきた写真の量はとんでもなく多い。これはなんとか整理してあげたい。X68000しかないっしょ。このグラフィックの性能ですもんね。写真のデータベースを作りたいわけです。本気でやれば光磁気が必要になるだろうけど，そこまで手が回らないからプロになるまでにその骨格だけでも作りたいと思っています。きっとわからないことだらけだと思いますので，よろしくお願いします。

森本　雄一郎(21)東京都

▲石田　伯仁　神奈川県
やっぱりMZのエプロンを着ているのがお姉さんで，Xのコートを着ているのが妹かな？　さぞ仲の良い姉妹でしょうね。

よきにはからえ。

◆12月20日午前１時23分。実はこの１年間成長させたOSを初期化してしまった(10Mバイト)。本当はバックアップを取るつもりが誤ってフォーマットしてしまった。気づいたときにはもう40パーセント初期化をしていた。ああ，ZMUSICが，OPMDが，その他いろいろ……。眠いときにシステムをいじるのはやめよう。

大森　和宏(16)千葉県

どうやって間違ったんだろう。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲　間

★クラブ「OREGA」では，年間８から12回の会報発行のほかに，共同開発，イベント参加，会員集会などを企画しています。会報はプログラミング講座，ハードウェア講座（「X68000ディスプレイにメガドライブをつなぐ」など），テクニカル情報，読書案内，エッセイ，漫画，イラストなどもりだくさんです。入会ご希望の方は，入会案内書をお送りしますので62円切手２枚を同封のうえ，郵便番号，住所，氏名を明記して下記までお送りください。〒910　福井県福井市文京4-2-22　サンパレット文京705　新海敏之方「OREGA入会希望Ｘ」係

★サークル「ごんべえ」ではただいまX68000ユーザーを大募集中です。活動内容は月１回の会報の発行や情報交換などです。大いなる野望を持つこのサークルに入会してみませんか？　興味を持った方は62円切手同封のうえ下記にご連絡を。〒946　新潟県北魚沼郡小出町四日町697　滝沢充浩(18)

★いままでＸ１を中心に活動してきました「VIP ROOM」ですが，Ｘ１ユーザーの減少，Ｘ68000ユーザーの超増加現象に伴い，Ｘ68000を中心としたサークル「VIP ROOM DELUXE」として再スタートすることになりました。活動内容は音楽を中心としたディスクマガジンの発行です。目標会員数は５億人。我こそはと思う方は62円切手同封のうえ，下記の住所までご連絡ください。〒199-02　神奈川県津久井郡藤野町小渕1740-7　佐々木孝司

売ります

★X68000用プリンタ「CZ-8PK9」，ほとんど使用していないものを送料込みで３万円で。箱，マニュアルあり。連絡は往復ハガキで。〒416　静岡県富士市蓼原720-1　渡辺直樹(15)

買います

★X68000用数値演算プロセッサボード「CZ-6BP1」を３万円（送料込み）で。連絡は往復ハガキで。〒731-42　広島県安芸郡熊野町柿迫49　長石裕行(21)

★X68000で使用できるハードディスクを。40Mバイトは４万円程度，80Mバイトは６万円程度で。完動品，取扱説明書，付属品があるものを。安価優先。連絡は希望価格などを書いて往復ハガキで。〒366　埼玉県深谷市藤ノ木65-1　加藤勲(20)

★OS-9/Ｘ68000用テクニカルマニュアル，ディスク付きで１万円から１万５千円で譲ってください。〒981　宮城県仙台市青葉区北山2-9-31　佐藤荘太郎(40)

★MIDI音源「CM-32L」を３万円程度で。また，Ｘ68000で使用できる40Ｍバイトのハードディスクを４万円程度で（CZ-64Hでも可）。完動品，付属品は最低条件。希望価格を明記のうえ，往復ハガキでお願いします。〒230　神奈川県横浜市鶴見区汐入町1-26　新井健史(19)

バックナンバー

★1987年９月号を送料込み1,000円で買います。切り抜きは不可。連絡はハガキでお願いします。〒491　愛知県一宮市大字丹羽字虚空蔵809-1　今枝務(20)

★1987年３月号または1989年２月号を1,500円（送料込み）で買います。S-OS“SWORD”に関する記事のところが完全であれば切り抜きしてあっても可です。連絡は往復ハガキでお願いします。〒079-03　北海道空知郡奈井江町北町４区　市村将尚

★1989年5,6,7月号を送料込み各1,500円で買います。切り抜き不可。連絡はハガキでお願いします。〒289-13　千葉県山武郡成東町成東2470　安井忍(23)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は1月号の内容に関するレポートです。

●今月の表紙は中華風でしょうか？　なかなかだと思います。先月号の表紙のゾウさんはちょっとびっくりしました……。もっとディスクつきというのを大きく表記してもよかったと思う。Oh!PCやI/Oとかだって，ディスクつきのときはデカデカと「ディスクつき」てな感じですからね。あとディスクに何が入っているかをもっとわかりやすく書いたほうがいいでしょう。中を見たら「付録5″2HDディスク……」の右どなりのやつがディスクの内容だとわかったけれど，表紙だけ見てもわからない。このへんは改善の余地あり。

畑　剛志(19) X1turboZII,MSX/2, JR-100　北海道

●今回の特集「急接近！　SX-WINDOW」は「初めの1歩」といった感じでしたが，それでよかったと思います。なぜなら，ほとんどの人がSX-WINDOWを使ったことはあっても，SX-WINDOW上のプログラムを組んだことがないのですから。SX-WINDOWに限らずウィンドウシステム上のプログラムを組んだことのある人もひと握りではないかと思います。そのような人たちにとり，この記事は感覚的にウィンドウ上のプログラムがどのような働きをするかということを知るうえで役立ったのではないでしょうか。

高橋　毅(19) X68000 PRO,MSX 埼玉県

●「光磁気ディスクシステムCZ-6MO1」の紹介記事について。できれば実際の使用例なんかを含めて，もう少し詳しい記事にしてくれたらうれしかったのですが。でも，よくまとまっていてわかりやすかったです。ただ説明が少々急ぎ足な気がするので，もう1ページくらい余分に使って説明してくれればよかったと思います。X68000自体のハードは変わらないのでこういった新しいデバイスの「分解体験記」でもしてもらうとうれしい。私はこんな大容量は使わないので買うことはないと思いますが，しいて購入条件あげるなら，コンピュータの主記憶がひと桁増えたような場合でしょう。個人的にはハードディスクは好きじゃないので，将来フロッピーディスクが容量的に使いものにならなくなったら購入するのでは，と思います。P.S.「磁気工学」の試験でしっかり光磁気ディスクのことを書きましたよ。

泉　昭彦(20)　X1turbo,PC-E500　東京都

●とうとうMOドライブが発売されてしまいましたが，45万円という価格は一般大衆には高いと思います。しかし，MOドライブを使いまくる人には決して高くはないと思います。私もX68000とお金さえあれば絶対にほしいですね。購入する条件としては光磁気ディスクカートリッジを3万円から1万円くらいに値下げして，MOドライブのデザインをマンハッタンシェイプに合わせたデザインにしてくれたらいいなと思います。

船越　直弥(18)　MZ-1500　北海道

●しかしまあ，今月はSX-WINDOW一色でしたね。メインメモリ1Mバイトだったころは，少ないメモリをいかに効率よく使おうかとバッチファイルを書きまくったものです。2Mバイトじゃ PASCALコンパイラ（私はPASCALのほうが好きなんです）と辞書を一緒にRAMディスクにおけるし……。きっと，そのうちまた「2Mバイトじゃ足りない！」なんて言い出すんでしょうね。ところで，HASHっていいですよね。きれいですよ。うん，とっても自然。いままで私ってば「レイトレーシングってきれいだけど，いまいちキカイっぽい」って思っていたんです。それでHASHの絵を見て「あーっ！」，私の求めていたものはこれだ！　と思ったんです。影ひとつであんなにも変わるものなんですね。しみじみ。

安井　百合江(16)　X68000 PRO　愛知県

●「清水和人流プログラミング道場」について。まさに，清水さんの思想を知る絶好の機会です。世の中，口だけの人が多いものですが，清水さんの場合，ちゃんとプログラムを提示してくれています。これは説得力がありますよ。読んでいて，実際に「俺もひとつやってみるか！」と奮起することはまずないのですが，読んでいるだけで楽しくなるので，この企画は好きです。

浅野　憲(19)　X1turboIII,X1Fmodel20,MZ-80C,FM-77L2, M5Jr.,PC-6001, PC-1245　大阪府

●「S-OS用COLUMNS」について。年末はこれにとりつかれて，勉強があまりできなくなってしまった。これは自業自得というものだが，なるほどよくできたゲームだと思った。なにより，プログラムが短くて打ち込むのが楽でよかった。付録ディスクは少々ものたりないような気もするが，前回が普通じゃなかったことは明白だから，これでいいと思う。SX-WINDOWの資料が入っていたのがいい。

土谷　興正(19)　X1，MSX2　兵庫県

ごめんなさいのコーナー

2月号　THE SOFTOUCH　「KLAX」

ゲームレビューの文中でアーケード版の日本での版権元がSEGAとなっていましたが，正しくはナムコです。お詫びいたします。

2月号　LIVE in '91

Misty Blueより「オープニングテーマ曲」は，「ファッションショーのシーンのBGM」の間違いでした。

1月号　DOCTOR2.X（謹賀新年PRO-68K）

先月号でお知らせしたようにチェック項目に不備がありました。DOCTOR2.Xを直接書き換えることでデバッグを行いますので，DOCTOR2.Xをカレントディレクトリにおき，次のBASICプログラムを実行してください。

```
10 fp=fopen("doctor2.x","rw")
20 fseek(fp,&H01FA,0) : fputc(&H66,fp)
30 fseek(fp,&H1538,0) : fputc(&H87,fp)
40 fseek(fp,&H1539,0) : fputc(&H0A,fp)
50 fseek(fp,&H153A,0) : fputc(&HE0,fp)
60 fseek(fp,&H153B,0) : fputc(&H4B,fp)
70 fclose(fp)
```

1990年10月号　INTEGRAL X1

リスト1　XLOAD.X1の1960行で，

GOTO i,0

というのがありますが，これは，

PALET i,0

の間違いです。申し訳ありませんでした。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

付録ディスクは5月号に決定 中身は大丈夫か?

▼まずは，いきなりですがお詫びです。このコーナーで次の付録ディスクは4月号の予定ですとお知らせしたわけですが，制作進行上の理由でディスクは5月号で付けることになりました。ごめんなさい。詳しい内容に関しては「ヒ・ミ・ツ」ですが，「ディスクのオープニングに使ってください」とオリジナル曲に取り組む西川善司氏をはじめスタッフ一同制作に励んでおります。また，とある超有名メーカーの協力で素敵なプレゼントが収録されるといったウワサも進行しております。お楽しみに。

▼今回の音楽特集はひさしぶりにMIDIを取り上げてみました。MIDIといえば一般的にいってコンピュータとは切り離された世界，すなわち純粋に音楽の世界で活発に利用されているのが実情です。それはもちろん音楽家にとってはあるべき状況なのですが，Oh!Xではそういった MIDIもパーソナルコンピュータの世界の一部として考えてみたいと思います。1月号の付録ディスク(謹賀新年PRO-68K)で配布したMUSICDRVは，そのための強い味方でもあるわけです。今回はMIDIの入門に加えてドライバ活用のアプローチを試みてみました。ぜひとも皆さんの手でパソコンとMIDIの関係を発展させてみてください。

▼さて今年も5月号で読者特集「言わせてくれなくちゃだワ」を開催いたします。閉じ込みのアンケート用紙は皆さんの声を誌面に反映させるための貴重な資料となりますのでぜひともご協力ください。特に最後の8番は重要ですので皆さんが伝えたいことをどんどん書いて送ってください。面白い意見やエピソードは「言わせて〜」の誌面で発表させていただきます。また，イラストも大々的に募集します(恒例のイラスト大賞の発表もあります)。なお，5月号は付録ディスクの関係で本の納品日が早くなりますので，アンケートの締め切りは3月15日，イラストの締め切りは3月22日とさせていただきます。ぜひともご協力ください。お待ちしています。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ(マシン語の場合)に，参考文献を明記し，プログラムをセーブしたテープ(ディスケット)を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋳㋮㋠」係

S H I F T ・ B R E A K

▶ついにアセンブラに手を出し始めた。やっぱり最初はBから，そしてCへとステップアップしていこうとしたが，あえなく撃沈。それが最近になって，いやでもアセンブラを使わざるを得ない状況になり，しぶしぶ使い始めたが不思議なことになんとかなっている。そして，いまさらながら68000ってすごいなあと感心している時代遅れな僕であった。(純)

▶スタッフ仲間でスキーに行ってきました。スタッドレスを履いて山岳路を行きましたが，雪道の怖さを知らない東京人のおバカさんたちは「ラリー走行みたいだね」とテールスライドしながら走りまくり，しまいにガードレールに衝突してしまいましたとさ。もう少しでOh!Xの人手不足をいっそう深刻にするところだった。(浦)

▶わかつきめぐみさんの漫画について教えてくださった皆さん，ありがとうございました。たくさんの葉書をいただきました。が，編集さんには「ちょっと調べりゃわかるようなことをわざわざ読者に聞くんじゃないっ!」と怒られてしまったのです。だって私が本屋の少女まんがコーナーに行って調べてると，まわりの人がひそひそと……，ううっ。(で)

▶ついにというか，やっと出た感じの新型シルビア。噂どおりSR20DEエンジンを搭載してきたが，やはり注目はインタークーラターボ搭載のK'sだ。馬力で30馬力，トルクで5kg-mアップは大きな魅力だ。人気車種だけあって，外観の変更はちょこっとだけ。う〜ん。MC前のリアウイングのほうが好きだったよ，わたしゃ。(H.K.)

▶先日オーダインのROMを買ったのだが，値段が「インストラクション・カード付き」と「無し」で分かれていた。そいで私は「付き」のほうを注文したのだが送られてきた箱の中には入ってなかった。いったいどういうことかと尋ねたら「手に入り次第送る」とのこと。おいおい，もう3カ月も待ってるんだぞトライさん!(善)

▶誘惑に負けてAMIGAを買ってしまった。お勧めなのがS端子つきのテレビ。電波新聞社の変換ユニットで，RGBモニタに匹敵する画質が得られた。でも原稿を書く間に遊ぶだけではない。今日RS-232Cケーブルで X68000と接続した。対戦ポピュラスをするためではなく，ターミナルデバッガとして働いてもらうためなのだ。うん，けっこういい環境。(A.T.)

▶この春には32ビットのX68000が出ると期待している人，水をさすようで悪いけどまだその可能性はほとんどなし。まあ，目標の8割は達成したとか，短期間で終わるとかいった話を鵜呑みにしたり希望的に解釈する人がこんなにも多かったわけだから，ネットがあらぬ噂で賑わうのもむりはないですが。なんにつけても冷静な判断力が欲しいですね。(S)

▶お正月のテレビ特番を観て，ちびまるこちゃんが好きになった。単行本も全部揃えたし映画も観てきた。ほのぼのとした情景が自分の古きよき時代と重ね合わせられるところがよい。しかし，まるちゃんと共通体験を持ってないはずの今の小中学生はどこを面白いと思うのだろうか。それとも少年(少女)時代は今も昔も変わらないのだろうか。(KO)

▶聞いた話によると，また中東で戦争が勃発したらしい。ひっきりなしに報道がなされているが，周りが平和すぎて(本当にいいことである)，あまりピンとこない。心配しているのは唯一，あのあたりの遺跡が無事なうちに戦争が終わってくれるかどうかということだけ。こんなふうに考えているのって，自分勝手なんだろうか?(A)

▶1月中にあったヘンなナンパ。その1「姉ちゃん，イモ食べない?」話好きの焼きイモ屋さん，結局2本タダで食べた。その2「よろしくお願いしまっす」路上で突然花束を差し出した大学生，ねるとんの見過ぎか? その3「モデルやりません?」このパターンは結構多いが，仕事そっちのけで飲んでるヤツは初めてだ。ああ，冬っていい。(E.O.)

▶「起」は始める。「承」は受け継ぐ。「転」はコケる……。ということで，付録ディスクは5月号。調べる暇のない昨年からの素朴な疑問。映画「ネバーエンディングストーリー第2章」に原作者はいるのだろうか? 前作があんなだったのにどうしてエンデは続編を許したのだろうか? うーん，わからない。(U)

▶お住まいの物件の所有権に変更があり，……てな調子で家賃の振込先を変えるという通知がきた。この手の詐欺は結構よくあるらしい。そうそう騙される人もいないと思うが，わざと1万円ぐらい振り込んで(今はこれだけしかなくて……と手紙を添えるのがコツだそうだ)詐欺罪を成立させる世のためだったかもしれない。皆さんもお気をつけて。(T)

microOdyssey

小学生の頃から図画工作とか，美術の時間というものがきらいだった。別に絵を描いたり，彫刻をしたりするのがきらいだったわけではない。そもそも創作するのに時間制限があるということがいやだったのだ。また，評価されるということも。

なにをするにも，手をつけるまでにわりと時間がかかるほうだったので，期限の直前になって無理やり仕上げるということの繰り返し。そうやって作り上げたものに満足するわけがない。出来上がった作品は自己嫌悪を感じるようなものばかりだった。

まあ，それならば家で趣味としてゆっくりと絵を描いたりすればよかったのだが，そういうこともしなかった。絵がうまいとか好きだとかいうわけではなかったから。

しかし，最近になって絵を描くということに興味が出てきた。絵を描くといっても紙の上ではなくディスプレイの上に，しかも，動くやつをである。動かすということで絵がうまくなくても，なんとかごまかせるし，既成のイメージを加工するだけでも面白い。

また，コンピュータ上に絵を描くということには，「準備に手がかからない」，「何度も描き直しができる」，「きれいな線や円が描ける」，「いくつでも描きかけの絵を保存しておける」という利点もあるので，絵の下手なものにとってはありがたいかぎりである。

使っているのは「DELUXE PAINT III」というソフト。もちろん，AMIGA用のソフトである(毎度まいどAMIGAの話ばかりで申し訳ない)。「DELUXE PAINT II」がPC-9801用に移植されるという話であるから，知っている人も多いかもしれない。

このソフトは評判どおり本当に使いやすく，お絵描きソフトというものを使ったことのないのにもかかわらず，快適に操作できる。もっとも，ほかのお絵描きソフトを使ったことがないので，どこがどう優れているということはわからないし，機能を自慢するつもりもない。いいたいのはとりあえず描いていて楽しいというだけである。

しかし，このソフトもアニメーション機能がなかったらこんなには楽しく感じられなかっただろう。静止画を描いてもこちらの絵心のなさから絶望するだけだったろうし。

なんでもない絵でも動かすことで，見るものに与える印象はかなり変わる。ちょいちょいちょいと描いたものでも，変わった動きさえすれば自分でも満足できるし，人にも結構感心してもらえる。

かといって，あまり長いものとか，ストーリー性のあるものを作るという気はない。作っていきたいのは動きのあるアートとでもいうべきもの。うまくたとえることができないが，「笑うモナリザ」とか「落穂拾いをする，落穂拾いの人々」といったようなものだろうか（かなり悪いたとえだなあ)。ともかく，未来の美術館に飾ってありそうな，さまになる「動く絵画」が描ければ満足である。

絵が下手でもごまかせるとはいえ，やはりうまいにこしたことはない。「絵画教室にでも通えばうまくなるかなあ」と藁にもすがるような気持ちでいる，今日この頃である。 (A)

1991年4月号3月18日(月)発売

特集 GAMING INTERFACE
発表！ 1990年度GAME OF THE YEAR
第3回アマチュアCGAコンテスト結果発表
新連載
泉大介「吾輩はX68000である」
中森章「SX関係資料解読術」

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3200)9185
	渋谷	大盛堂書店	03(3463)0511
	池袋	リブロ池袋店	03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコン∑上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 3月号
■1991年3月1日発行 定価560円(本体544円)
■発行人 孫 正義
■編集人 橋本五郎
■発売元 ソフトバンク株式会社
■出版事業部 〒108 東京都港区高輪2-19-13 NS高輪ビル
Oh!X編集部 ☎03(5488)1309
出版営業部 ☎03(5488)1360 FAX 03(5488)1364
広告センター ☎03(3297)0181
■印刷 凸版印刷株式会社

バックナンバー案内

ここには1990年3月号から1991年2月号までをご紹介しました。現在1990年9～12, 1991年1～2月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については，172ページを参照してください。

1990

3月号（品切れ）
特集　MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
●X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90　パワードリフト/スキーム/となりのトトロ
THE SOFTOUCH　ナイトアームズ/斬/ダンジョンマスター
全機種共通システム　超多機能アセンブラOHM-Z80

4月号（品切れ）
特集　ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040，ついに登場
LIVE in '90　バーニングフォース（OPMD対応）
THE SOFTOUCH　The Fille Professor/HOST PRO-68K
全機種共通システム　ファジィコンピュータシミュレータI-MY

5月号（品切れ）
特集　BASICプログラミング
第5回　言わせてくれなくちゃだワ
連載　ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90　TURBO OUTRUN
THE SOFTOUCH　天下統一/ポピュラス/Hyperword
全機種共通システム　インタプリタ言語STACK

6月号（品切れ）
特集　創刊8周年記念PRO-68K（付録5″2HD）
Oh!Xアンケート結果大分析大会
連載　ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90　ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH　三国志II/FAR SIDE MOON/グラナダ
全機種共通システム　X68000用S-OS"SWORD"他

7月号（品切れ）
特集　マシン語への第一歩
X68000SUPER-HD試用レポート
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1――ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90　夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH　サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム　リロケータブルアセンブラWZD

8月号（品切れ）
特集　ADVANCED 2D GRAPHICS
100号記念特別モニタプレゼント
連載　ショートプロぱーてぃ/Z80's Bar/INTEGRAL X1
X-BASIC調理実習/X68000マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●X68000用画像回転プログラム XROT0.X
LIVE in '90　OMENS OF LOVE/ENDLESS RAIN/ダートフォックス
THE SOFTOUCH　大航海時代/ウルティマV/プロミストランド
全機種共通システム　リンカWLK

9月号
特集1　日本語を処理するための序章
特集2　ADVANCED 2D GRAPHICS
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90　風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH　T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム　BILLIARDS

10月号
特集　電子音楽術入門
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語
LIVE in '90　Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH　ワールドコート/ルーンワース/闇の血族/提督の決断
全機種共通システム　ライブラリアンWLB

11月号
特集　理科系のGAME REVIEW
連載　Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL X1
荻窪圭の大人のためのX68000
LIVE in '90　ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL　ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム　スクリーンエディタEDC-T

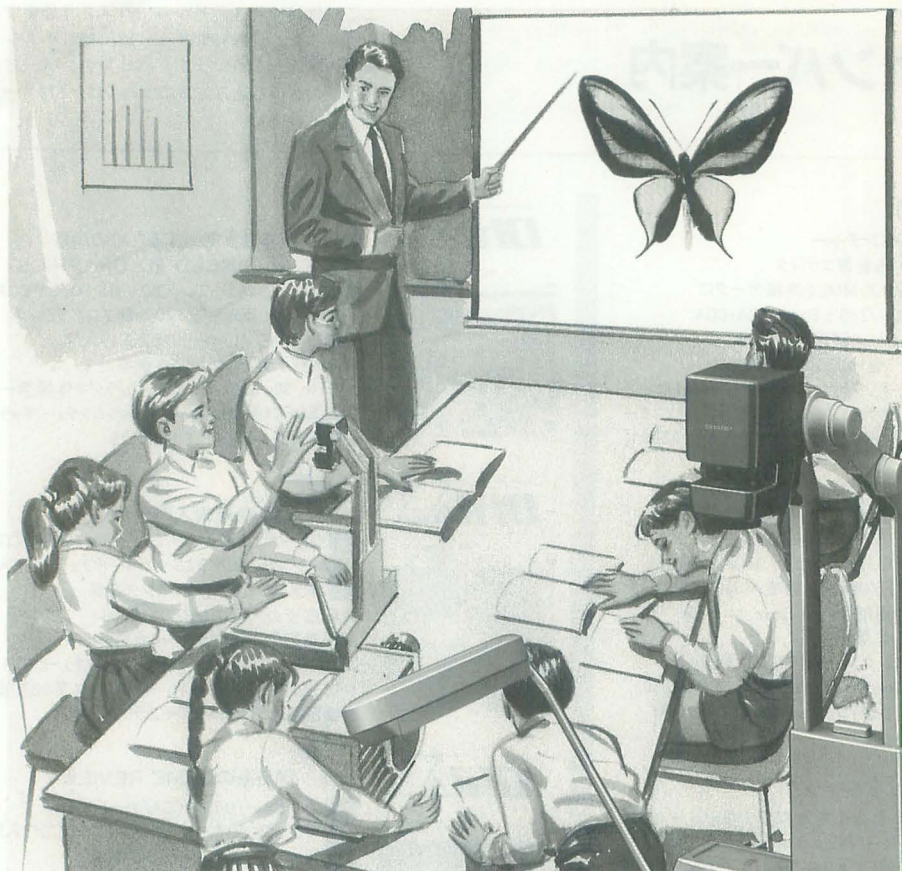
12月号
特集　XCのための傾向と対策
連載　X-BASICプログラミング調理実習/ハードウェア工作入門
マシン語プログラミング/ショートプロぱーてぃ/Z80's Bar
大人のためのX68000/ようこそここへC言語/INTEGRAL X1
●シミュレーションプログラミング入門
●特別企画アナログジョイスティックの製作
LIVE in '90　グラディウスIII/メタルサイト
THE SOFTOUCH SPECIAL　イメージファイト/ジェミニウイング/NAIOUS他
全機種共通システム　STACKコンパイラ

1991

1月号
特集　急接近！SX-WINDOW
特別付録　謹賀新年PRO-68K（5″2HD）
連載　ハードウェア工作入門/シミュレーションプログラミング入門
DōGA・CGA/ショートプロぱーてぃ/大人のためのX68000
PurePASCAL/清水和人流プログラミング道場/X-BASIC調理実習
LIVE in '91　めぞん一刻/涙で綴るパパへの手紙
THE SOFTOUCH　ソル・フィース/銀英伝II/続ダンジョン・マスター他
製品紹介　光磁気ディスクCZ-6MO1
全機種共通システム　ブロックアクションゲームCOLUMNS

2月号
特集1　グラフィックの"実験的"手法
特集2　SX-WINDOWプログラミング
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/INTEGRAL X1/ようこそここへC言語
●1990年度 GAME OF THE YEAR ノミネート発表
LIVE in '91　Misty Blue/スプーンおばさん
THE SOFTOUCH　栄冠は君に/KLAX/ダイナマイト・デューク他
全機種共通システム　ダイスゲームKISMET

SHARP

X68000シリーズ対応

シャープ「ビデオ レセプター」登場。

理解を深める美しい色再現性と高解像度、立体物や印刷物もそのまま映し出せる汎用性。この高画質がOHPにかわる新しいツールとして視聴覚教育を支援します。

標準価格398,000円(税別)

【美しく伝える】
RGB原色信号順次読み取り撮像方式による高性能カラービデオカメラを採用。75万画素(25万画素×RGB)の高解像度で文字や絵をくっきりと映し出す高画質、鮮やかな色再現性を実現しました。

【便利なコールバック機能】
撮像画面を記憶し、必要なときにすぐ呼び出せるコールバック機能を内蔵。原稿を取り替える手間が省けます。別売の増設メモリボードを装着すれば、本体内蔵のメモリ(1画面分)とあわせて合計4画面分が記憶できます。

【原稿を選ばない】
手持ちの資料、雑誌やカタログなどの印刷物、また教材としてよく使われる生物の標本などの立体物も、そのまま映し出せます。

【大画面の迫力】
液晶ビジョン(別売)との組み合わせて、最大100インチの大画面プレゼンテーションが可能となります。

【実用的な使いやすさ】
ビデオ/S映像/RGB出力端子を装備してテレビやパソコンディスプレイなどさまざまな表示装置に対応。また35㎜フィルムサイズからB4サイズまでスムーズに撮像できる8倍ズーム機能、写真のネガをそのまま撮像できるネガ・ポジ反転機能、視聴覚教室の音響設備を利用できるマイク入力/音声ライン出力端子を装備しています。

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券

満開の電子(でんこ)ちゃん

作：いわいいっぺい
え：岡村祭

丹　明彦（福井県）

物足りない毎日に刺激のほしい人、美しく楽しい絵や音楽を鑑賞したい人、便利なツールをすぐにでも使いたい人……すべてのX68Kユーザーに贈るディスクマガジン、それが電脳倶楽部です。ユーザーの活動の成果を凝縮した濃い内容。作者のこだわりが伝わってくるようなツール群はかゆいところに手が届くようです。資料的価値も高いし、ディスクメディアですので参照や再利用も簡単。僕もお世話になっています。リアルタイムな楽しみ方から血となり肉となる使い方までをサポートする電脳倶楽部を読んで幸せになりましょう。

2枚のボードが1枚になった

KGB-X68PRK

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2、CZ-6BE4、CZ-6BP1との混在が可能
- 複数枚のKGB-X68PRKの実装が可能
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能
- ジャンパの変更により数値演算プロセッサの1枚目、2枚目、未使用の選択が可能
- 1M、2M、3Mメモリモデルは購入後もメモリ増設が可能
- PRK-10、11、12、13、14にはデバイスドライバ(FLOAT3.X)が付属

真はKGB-X68PRK-14です

Z-600C、601C、611C、652C、653C、662C、663Cで御使用の際にはあらかじめ専用の1Mメモリ(CZ-6BE1、A、B等)でメインメモリを2Mバイト上にしておく必要があります。

製品価格一覧

製品	価格
KGB-X68PRK-00 (メモリ無し/数値演算プロセッサ無し)	¥34,000
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥58,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥74,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥98,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000
KGB-X68PRK-10 (メモリ無し/数値演算プロセッサ付き)	¥76,000
KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥96,000
KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥112,000
KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

PRK質問箱

購入後のメモリ増設はどうやるのでしょう?
ご購入後のPRKに対するメモリの増設は半田付け等の技術を要するため原則として当社に送り返していただき増設いたします。自分でメモリ増設をする場合は通信販売のみですが必要な部品の販売も致します。御希望の方はお問い合わせ下さい。

数値演算プロセッサにMC68882を使用することは可能ですか?
MC68882では動作しないソフトが存在するため使用できません。

「数値演算プロセッサのみ」や「プロセッサ無しメモリ無し」のPRKがほしいのですが?
PRK-10、PRK-00の型番で商品化しております。

最近PRKをスロットに挿入したが動作しないと言う御質問を良く受けますが、ほとんどの場合は差し込み不足が原因です。X68000のスロットは大変堅く裏蓋が閉まる状態でも差し込み不十分の場合があります。御注意ください。

おしらせ

バージョンアップサービス

★BASIC拡張関数パッケージ(B6-6306)
(C言語ライブラリー付き)

★C言語ライブラリー(B6-6305)
SHARP XC Ver.2に対応になりました。新バージョンでは従来のXFUNCLIB.Aの他に新たにXFUNCLIB.Lが追加されています。

★DISK CACHER
ハードディスクキャッシュの大幅な高速化が行なわれました。
HDISKCACHE.SYSのVer.2.00未満をお持ちの方が対照になります。

バージョンアップご希望の方は旧バージョンのディスクラベルと代金を同封して現金書留で通販部宛にお申し込みください。

B6-6306(拡張関数ライブラリー付き)	¥2000
B6-6305(C言語ライブラリー)	¥1500
B6-6304(ディスクキャッシャー)	¥1500

※送料、手数料、税込みの価格です。

充実のBASIC HOUSEソフトウェア&ハードウェア

製品	価格
速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
ォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
イソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥ 68,000
ンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800
速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
180CPUボードMach 180(KGB-CPXB) X68000	¥ 98,000
ーコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

ASIC拡張関数パッケージ(B6-6301)¥9,800　C言語ライブラリ(B6-6305)¥6,800
ィスクキャッシャー(B6-6304) ¥6,800　Toys & Tools (B6-6307)¥6,800
ASIC拡張関数パッケージC言語ライブラリ付(B6-6306) ¥14,800
イコンエディタ(B6-6303) ¥4,800　CP/M68Kエミュレータ(B6-6302)¥19,800

ビデオボードを外付けに!!
ビデオボードケース(KGB-BVBX)
大好評発売中 定価9,800円

SHARPより発売されているCZ-6BV1を外付けにするケースです。このケースの使用によりあなたのX68000のスロットが開放されます。

Human68k下のソフトのCRT出力を強制的に15kHZ出力にする(768×512モード除く)おまけユーティリティ付き

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部/マイコンショップ/通販部　宇都宮市竹林町503—1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所/マイコンショップ　大田原市美原1—13—4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

AIBIT
アイビット電子株式会社

SHARP
パソコン本体から周辺機器まで品数取り揃え
大特価セール実施中!!

型名	品名	正価	特価
PC-E500BL	ポケコン	28,800	19,500
PC-1600K	ポケコン	69,800	49,800
PC-1360K	ポケコン	36,800	32,800
PC-1360	ポケコン	29,800	19,800
PC-1262	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1280	ポケコン	24,800	19,600
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800
CE-203M	ポケコンRAM32K	32,000	7,000
CE-202M	ポケコンRAM16K	35,000	6,000
CE-201M	ポケコンRAM 8K	18,000	3,000
CE-1600M	ポケコンRAM32K	32,000	16,000
CE-1600F	ポケコンフロッピードライブ	39,800	34,800
CE-1600P	ポケコンプリンター	69,800	59,800
CE-1650F	ポケコンDISK	9,800	8,800
CE-161	ポケコンRAM16K	50,000	3,800
CE-1601M	ポケコンRAM64K	45,000	30,000
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-159	ポケコンRAM 8K	35,000	4,200
CE-140T	ポケコンRS-232Cコンバター	9,800	8,800
CE-140F	ポケコンフロッピディスク	49,800	44,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-120P	ポケコンプリンター	24,800	21,800
CE-124	ポケコンカセットインター	4,500	3,600
Z-VISIONplus	Z80シュミレータ デバッカー	59,800	51,000
UX-1	ホームコピーファクス	78,000	69,800
PA-9500	ハイパー電子手帳	48,000	特価
CZ-300F	X13"マイクロフロッピー	79,800	9,000
CZ-31FS	300F増設フロッピー	59,800	7,000
CZ-82F	CZ-802C増設フロッピー	59,800	6,000
CZ-501H	X1増設用ハードディスクユニット	258,000	60,000
CZ-503F	CZ-830増設ドライブ	49,800	30,000
CZ-520F	2HD/2DDミニフロッピードライブ	118,000	70,000
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIB	1M増設RAMボード	28,000	19,500
CZ-6BE1	1M増設RAMボード	35,000	29,500
CZ-6BE2	2M増設RAMボード	79,800	63,800
CZ-6BE4	4M増設RAMボード	138,000	110,400
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BF1	RS-232C増設ボード	49,800	42,300
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU	RRGBシステムチューナー	33,100	26,500
CZ-6BG1	X6800GPIBボード	59,800	50,000
CZ-6BC1	X6800FAXボード	79,800	67,800
CZ-6PV1	ビデオプリンター	198,000	158,000
CZ-6BV1	ビデオボード	21,000	16,800
CZ-822C	X1G MODEL30	118,000	39,800
CZ-820C	X1G MODEL10	69,800	16,800
CZ-888C	X1TURBO		
CZ-8BGR2	グラフィックボードX1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	X1 漢字ROM	19,800	16,800
CZ-8BM2	232Cマウスボード	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット		
CZ-8BV2	カラーイメージボート	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8TM1	X1ソフト付モデムユニット	29,800	5,000

型名	品名	正価	特価
CZ-8TM2	X1ソフト付モデムユニット	49,800	39,800
CZ-8EB3	拡張i/o box	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	
CZ-8PK7	24ドット80桁漢字プリンター	122,000	59,800
CZ-8PC5BK	48ドット熱転写カラー漢字プリンター	96,800	新発売
CZ-8BS1	X1FM音源ボード	23,800	19,800
CZ-8BK4	X1第2水準ROM	———	5,700
CZ-8NJ2	インテリジェントコントローラー	23,800	18,500
CZ-8NS1	カラーイメージスキャナー	188,000	149,000
AN-S100	アンプ付スピーカー	36,600	29,500
AN-X68	キーボードシリコンカバー	3,500	2,800
AN-X68PRO	キーボードシリコンカバー	3,500	2,800
AN-1508	ディスプレイ15P→8P変換ケーブル	———	1,600
AN-1506	ディスプレイ15P→6P変換ケーブル	———	1,600
HXD040	アイティム40Mハードディスク(ITM)	118,000	95,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	79,800
CU-14FD	カラーディスプレーアナログ0.31	74,800	49,800
MZ-1D10	12"モノクロディスプレー	41,800	25,000
MZ-1D17	15"CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000 プリンターI/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレルI/F	34,800	28,000
MZ-1E45	MZ6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレル I/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス 300BT	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500 QD用インターフェイス	9,800	3,000
MZ-1M01	MZ-2000/2200 16ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	45,000
MZ-80P4B	136桁ドットプリンター	———	48,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P28	ドットプリンター漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター136桁	168,000	134,400
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500 増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-28611M増設RAM	45,000	15,000
MZ-1R35	MZ-28611M増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R27A	MZ-2500VRAM	13,000	10,000
MZ-1R21	漢字ROM	38,000	13,000
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000

型名	品名	正価	特価
MZ-1R32	MZ-6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
MZ-1R29	MZ-1P22 増設RAM17-22	32,000	12,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1U09	MZ-2500 拡張ボード	9,000	7,200
MZ-1V01	パソコン FAX	278,000	85,000
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z016	MZ-5500 附属	———	5,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属	———	5,000
MZ-2Z013	MZ-5500 MSDOS	25,000	20,000
MZ-4Z001	MZ-5500 IBM変換	30,000	8,000
MZ-5521	本体	388,000	55,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト	———	3,500
MZ-6F03	ブランク QD DISK	450	400
MZ-6P18	MZ-1P18,28カットシートフィーダー	60,000	35,000
MZ-6P29	MZ-1P29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,000	2,700
MZ-6Z22	MZ-6500(50)CP/M86BASIC-3	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8B104	MZ200/2200 GP IBインターフェイス	45,000	18,000
MZ-8BC01	MZ200/2200 GP IBケーブル	18,000	8,000
UE-1U01	X286L スロットBOX	5,000	4,000
UE-1R02	4M RAMボード	300,000	240,000
UE-1R06	辞書ROMボート	32,800	25,600
UE-1R01	2M RAMボード	160,000	128,000
UE-1R05	拡張グラフィックボード	92,000	55,000
UE-1R03	1M RAMボード	100,000	80,000
UE-1R04	2M RAMボード	180,000	144,000
UE-1P03	80桁漢字プリンタ	———	特価
UE-1P04	136桁漢字プリンタ	———	特価
UE-1P05	136桁漢字水平プリンタ	———	特価
UE-1P02	高速136桁漢字プリンタ	550,000	440,000
UE-1P01	136桁漢字プリンタ	268,000	214,400
UE-1E04	S-RNインターフェイスカード	70,000	56,000
UE-1E02	AX286L ICカードI	45,000	36,000
UE-1E03	5"FDインターフェイスカード	28,000	22,400
UE-1D03	15インチカラーディスプレイ	123,000	98,400
UE-1D02	14インチカラーディスプレイ	158,000	126,400
IO-735X	カラープリンター	248,000	195,000
BF-68PRO	フィルター	19,800	16,800
X6800キーボード延長ケーブル(1.5m)		2,500	2,000
ディスプレーケーブルアナログ15P(3m)		5,000	4,000
ディスプレーケーブルアナログ15P(1.5m)		4,300	3,500

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

FM TOWNS/FM NOTE/東芝ダイナブック、周辺機器も取扱っております。

ACCESS

X1エミュレータ

好評発売中

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3〜5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

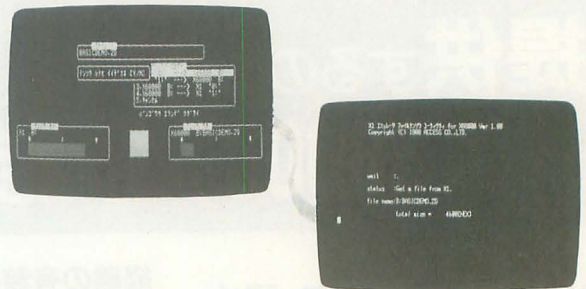
X1ディスク⟷X68000 Human68k(5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC:CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?

A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?

A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?

A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?

A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?

A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

*タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。

*一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

*この商品価格には消費税は含まれておりません。

*CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。

*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス 〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(3233)0200(代) FAX.03(3291)7019

T4910217903567 雑誌 02179-3